

Gentran Integration Suite

Services and Adapters

Version 4.3



Contents

Introduction	8
AFT Route Progress Event Reporting Service	9
Alert Service	12
Archive Business Process Service	19
Attachment Parsing Service	21
Auto Terminate Service	24
B2B Lookup System Service	32
B2B Mail Client Adapter	34
B2B Request/Response System Service	40
B2B Send System Service	42
B2B SMTP Client Adapter	51
Backup Business Process Service	52
Batch Processor Service	56
BEA Tuxedo Adapters	61
Begin Transaction Service	70
BP Exception Service	75
BP Fault Log Adapter	77
BP Fault XML Log adapter	80
BP Metadata Info Service	83
BP Report Service	87
Cache Refresh Service	89
CDATA Conversion Service	93
Certificate Revocation List (CRL) Processing Service	96
Check Expire Service	100
Command Line Adapter	104
Command Line Adapter 2	118
Compression Service	132
Connect:Direct Requester Adapter	137
Connect:Direct Requester Select Process Service	140
Connect:Direct Requester Select Statistics Service	150
Connect:Direct Requester Submit Process Service	159
Connect:Direct Server Adapter (Build 4300 - Build 4308)	166
Connect:Direct Server Adapter (Build 4309 or higher)	178
Connect:Direct Server Begin Session Service	190
Connect:Direct Server BP Response Service	195
Connect:Direct Server CopyFrom Service	198
Connect:Direct Server CopyTo Service	207
Connect:Direct Server End Session Service	217
Connect:Direct Server Run Job Service	222

Connect:Direct Server Run Task Service	227
Connect:Direct Server Submit Service	233
Connect:Enterprise UNIX Server Adapter	238
Connect:Enterprise UNIX Server Add Service	248
Connect:Enterprise UNIX Server Batch Status Service	254
Connect:Enterprise UNIX Server Begin Session Service	260
Connect:Enterprise UNIX Server End Session Service	264
Connect:Enterprise UNIX Server Extract Service	268
Connect:Enterprise UNIX Server Log Service	274
CPA Lookup Service	278
Cryptographic Message Service (Build 4315 or higher)	280
Data Sweeper Service (Build 4319 or higher)	293
DB Monitor Service (Build 4300 - Build 4321)	297
DB Monitor Service (Build 4322 or higher)	304
Document Keyword Replace Service	311
Document Tracking Service	320
Document XPath Replace Service	324
Dynamic Services	329
E-5 2000 Adapter	346
ebXML BPSS Correlation Service	361
ebXML BPSS Doc Parsing Service	367
ebXML Business Service Handler (BSI) service	370
ebXML CPA Lookup Service	373
ebXML Lookup Service	376
ebXML Manifest and Payload Service	379
ebXML Profile Service	381
ebXML Request Response Service	383
ebXML Validation Service	385
ebXML XML Digital Signature Service	387
EJB Adapter	396
Encoding Conversion Service	404
End Transaction Service	406
Execution Control Service	411
Export Service	414
File System Adapter (Build 4300 - Build 4324)	427
File System Adapter (Build 4325 or higher)	445
For Each Document Service	463
FTP Client Adapter	472
FTP Client Begin Session Service	479
FTP Client CD Service	487
FTP Client DELETE Service	492
FTP Client End Session Service	496
FTP Client GET Service	500
FTP Client LIST Service	508
FTP Client MOVE Service	513
FTP Client PUT Service	518
FTP Client PWD Service	526
FTP Client QUOTE Service	530
FTP Client SITE Service	535
FTP Reverse Proxy Adapter	539
FTP Server Adapter (Build 4300 - Build 4308)	540

FTP Server Adapter (Build 4309 - Build 4323)	548
FTP Server Adapter (Build 4324 or higher)	556
GS:Unix Purge Process Service	564
Gentran:Server Windows Adapter	568
Get Document Info Service	574
GXS ICS FTP Adapter	583
HTTP Client Adapter	590
HTTP Client Begin Session Service	596
HTTP Client End Session Service	603
HTTP Client GET Service	606
HTTP Client Method Service	615
HTTP Client POST Service	625
HTTP Respond Service	635
HTTP Reverse Proxy Adapter	638
HTTP Server Adapter	639
Human Interaction Document Loader Service	645
Human Interaction Event Service	652
Human Interaction Query Service	666
Human Interaction XForms Service	676
IBM Information Exchange FTP (IBM IE FTP) Adapter	689
Import Service (Build 4300 - Build 4321)	694
Import Service (Build 4322 or higher)	700
Index Business Process Service	706
Instant Messaging Adapter Suite	708
Invoke Sub-Process Service	720
iWay Adapter	728
Java Database Connectivity (JDBC) Adapter	733
JCA Listener Service and GIS Resource Adapter	749
JMS Queue Adapter (Build 4300 - Build 4317)	757
JMS Queue Adapter (Build 4318 or higher)	766
JMS Topic Adapter (Build 4300 - Build 4317)	775
JMS Topic Adapter (Build 4318 or higher)	784
Lightweight Directory Access Protocol (LDAP) Adapter	793
Lightweight Java Database Connectivity (JDBC) Adapter (Build 4300 - Build 4320)	802
Lightweight Java Database Connectivity (JDBC) Adapter (Build 4321 or higher)	838
Lock Service (Build 4300 - Build 4317)	875
Lock Service (Build 4318 - Build 4321)	881
Lock Service (Build 4322 or higher)	887
Mail Mime Service	895
Mailbox Add Service	903
Mailbox Correlate Document Service	907
Mailbox Delete Mailbox Service	909
Mailbox Delete Service	914
Mailbox Evaluate All Automatic Routing Rules Service	919
Mailbox Evaluate Routing Rule Service	922
Mailbox Extract Abort Service	926
Mailbox Extract Begin Service	930
Mailbox Extract Commit Service	934
Mailbox List Service	937
Mailbox Query Service	939
Mailbox Scheduled Delete Service	945

Mailbox Update Service	949
Map Test Service	952
Mapped Extraction Service	955
MergeDocument Service	958
MIME Service	961
MQRFH2 Service	975
MSMQ Adapter (Build 4300 - Build 4317)	979
MSMQ Adapter (Build 4318 or higher)	985
MSMQ Send Service	992
Network Report Reconciliation Service	998
Obscure Data - Obscure Primary Document Service	1002
Obscure Data - Process Data Values Service	1005
Obscure Data - Reveal Primary Document Service	1010
OdetteFTP Adapter	1012
OdetteFTP Queue Handler Service	1026
OdetteFTP Scheduler Service	1032
Oftp Adapter	1034
Oracle AQ JMS Queue Adapter	1041
Oracle AQ JMS Topic Adapter	1045
Oracle E-Business Suite Adapter Configuration Service	1049
Oracle E-Business Suite Message Service	1054
Oracle E-Business Suite Adapter	1059
Adapter for PeopleSoft	1070
Adapter for PeopleSoft CRM CIC	1090
Timestamp Service for PeopleSoft	1102
PGP Package Service (Build 4300 - Build 4314)	1104
PGP Package Service (Build 4315 - Build 4321)	1117
PGP Package Service (Build 4322 or higher)	1130
PGP Unpackage Service (Build 4300 - Build 4321)	1143
PGP Unpackage Service (Build 4322 or higher)	1153
Pre 3.1 Self Registration Service	1163
Purge Business Process Linkage Service	1166
Purge Service	1167
Release Service	1171
Remote Method Invocation (RMI) Adapter	1178
Report Service	1185
Restore Business Process Service	1187
SAP Suite Adapter (Build 4300 - Build 4311)	1190
SAP Suite Adapter (Build 4312 or higher)	1263
SAP XI Adapter	1345
Script Adapter	1367
Self Registration Service	1372
SFTP Client Adapter	1377
SFTP Client Begin Session Service	1386
SFTP Client CD Service	1392
SFTP Client DELETE Service	1395
SFTP Client End Session Service	1398
SFTP Client GET Service	1401
SFTP Client LIST Service (Build 4300 - Build 4324)	1405
SFTP Client LIST Service (Build 4325 or higher)	1409
SFTP Client MOVE Service	1413

SFTP Client PUT Service	1416
SFTP Client PWD Service	1420
SFTP Server Adapter (Build 4300 - Build 4323)	1423
SFTP Server Adapter (Build 4324 or higher)	1430
Simple Network Management Protocol (SNMP) Trap Adapter	1437
SMTP Send Adapter	1446
SOA Inbound Message Processing Service (Build 4303 or higher)	1463
SOA Inbound Security Service (Build 4303 or higher)	1469
SOA Outbound Message Processing Service (Build 4303 or higher)	1475
SOA Outbound Security Service (Build 4303 or higher)	1481
SOAP Inbound Service	1488
SOAP Outbound Service	1501
Sleep Service	1506
Sterling Control Center Service	1509
Sterling Information Broker Adapter	1512
Straight Through Extraction Service	1518
Text Document Appender Service	1520
This Service	1522
TIBCO Rendezvous Adapter	1540
Timestamp Utility Service	1546
Timezone Offset Service	1548
Typing Service	1550
User Service	1553
Vitria (Businessware) Adapter	1556
Wait Notify Service	1562
Wait Service	1564
webMethods Adapter	1566
WebSphere MQ Adapter	1586
WebSphereMQ Suite Async Receiver Adapter (Build 4300 - Build 4306)	1604
WebSphereMQ Suite Async Receiver Adapter (Build 4307 or higher)	1615
WebSphereMQ Suite Backout Service	1627
WebSphereMQ Suite Close Queue Service	1631
WebSphereMQ Suite Close Session Service	1635
WebSphereMQ Suite Commit Service	1639
WebSphereMQ Suite Get Message Service	1643
WebSphereMQ Suite Open Queue Service	1652
WebSphere MQ Suite Open Session Service	1656
WebSphereMQ Suite Put Message Service	1662
WSDL Service	1671
XML Digital Signature Service (Build 4315 - Build 4318)	1674
XML Digital Signature Service (Build 4319 or higher)	1694
XML Encoder Service	1715
XML Encryption Service	1718
XML Validation Service	1722
XSLT Service (Build 4300 - Build 4320)	1726
XSLT Service (Build 4321 or higher)	1733
Yantra Adapters (Build 4300 - Build 4311)	1741
Yantra Adapters (Build 4312 or higher)	1747
Zengin TCP/IP Adapter	1751
ZlibInflate Service	1771

Introduction

This guide contains documentation for services and adapters provided with Gentran Integration Suite 4.3 only. Documentation limited to specific build numbers is identified in the topic title.

For information about services and adapters provided with related products such as Sterling Standards Library or Sterling e-Invoice Gateway, see the respective documentation for the product.

AFT Route Progress Event Reporting Service

The AFT Route Progress Event Reporting service reports events through the progress of a advanced file transfer route so that reports can be generated. The following table provides an overview of the AFT Route Progress Event Reporting service:

System Name	AFT Route Progress Event Reporting Service
Graphical Process Modeler (GPM) categories)	AFT
Description	Reports an AFT routing progress event
Business usage	Use this service to add custom reporting events regarding AFT routing progress to a business process to augment the installed capabilities of Application.
Usage example	When you build a custom AFT routing business process, you can use this service to report AFT routing progress events.
Preconfigured?	This service is preconfigured as part of the system installation.
Requires third party files?	No third party files are required.
Platform availability	All supported platforms
Related services	None
Application requirements	None
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	Used within the business process.
Business process context considerations	None
Returned status values	SUCCESS or ERROR. If the status is ERROR, the status report contains the error details.
Restrictions	None
Persistence level	Not applicable
Testing considerations	Not applicable

How the AFT Route Progress Event Reporting Service Works

The AFT Route Progress Event Reporting service reports AFT routing progress events when it is called in an AFT routing business process.

Implementing the AFT Route Progress Event Reporting Service

To implement the AFT Route Progress Event Reporting service, use it in a business process.

Configuring the AFT Route Progress Event Reporting Service

There are no configurable parameters for the AFT Route Progress Event Reporting service.

Business Process Example

The following example illustrates how the AFT Route Progress Event Reporting service can be used in a business process:

```
<process name="AFTRouteReportProgressEvent">
  <sequence>
    <operation name="GetWorkflowId">
      <participant name="This" />
      <output message="thisRequest"/>
      <input message="thisResponse">
        <assign to="AFTRouteWorkflowId"
          from="this/node()/text()" />
      </input>
    </operation>
    <operation name="AFT Route Progress Event">
      <participant name="AFTRouteProgressEventService"/>
      <output message="ProgressBeginRequest">
        <assign to="AFTRouteId" from="'2000'" />
        <assign to="AFTRouteEventId" from="'AFT_8199'"/>
        <assign to="AFTRouteEventMessageParameters"
          from="'AFTRouteTest:AFTRouteTestTransportBP'" />
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Parameters Passed From Service to Business Process

There are no parameters passed from the service to the business process.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the AFT Route Progress Event Reporting service:

Parameter	Description
AFTRouteId	The data flow ID for this AFT route. Required.
AFTRouteWorkFlowId	The workflow ID for this AFT route. Required.
AFTRouteEventID	The ID of the progress event to report. Required.
AFTRouteEventMessage Parameters	A colon-delimited string containing the parameters for the event being reported. Optional.

Alert Service

The Alert service logs errors to an independent log file when business processes have errors. The following table provides an overview of the Alert service:

System name	Alert Service
Graphical Process Modeler (GPM) categories	All Services, System
Description	The Alert service runs as a scheduled service, independent of any specific business process. If an error occurs and is not handled by the business process, the Alert service detects the error and reports it, regardless of the business process that generated the error.
Business usage	Alert service not only provides an optimal method to monitor and notify Application errors in a production environment, also provides flexibility to track specific business processes.
Usage example	The Alert service is an independent business process. It does not need any input and does not work in conjunction with any other services. You can run the Alert service anytime you want to track failed business processes, and the Alert service sends notification on any error or warning that happens in any business process.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	Not applicable
Business process context considerations	None
Returned status values	None
Restrictions	None
Persistence level	None
Testing considerations	None

How the Alert Service Works

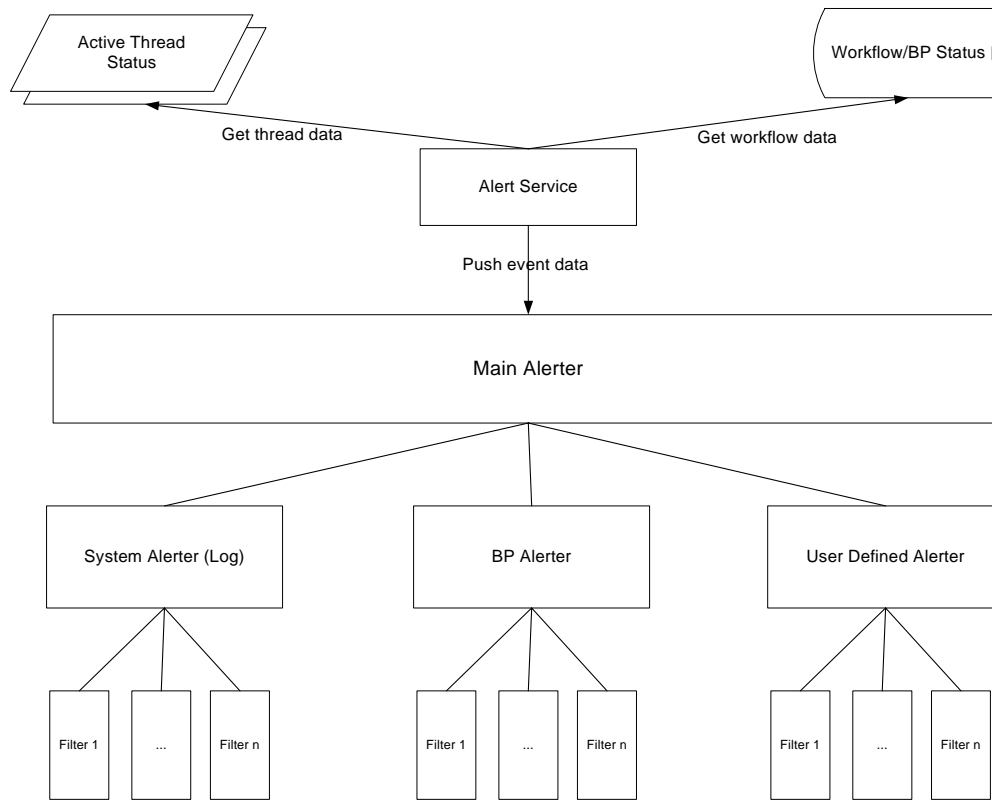
The Alert service runs as a scheduled service, independent of any specific business process. If an error occurs and is not handled by the business process, the Alert service detects the error and reports it, regardless of the business process that generated the error.

If an error occurs in a business process, the Alert service logs notification failures to an independent log file. An external business process can monitor this file for growth to determine whether the Alert service subsystem fails.

The Alert service consists of:

- ◆ A subsystem controller—the Main Alerter
- ◆ Individual instances of alerter and filter components
- ◆ A Application Alert service
- ◆ A default Application Alert Notification business process

The following figure shows how the components of the Alert service work together:



Main Alerter

The *Main Alerter* receives event data and delegates the responsibility for notifying the appropriate alerter instances. Each event has a type depending on the data that produced it. All alerters receive the event. The Main Alerter uses filters to determine whether the event needs to be reported. Each alert instance in turn calls the filter method on each of its configured filter instances to determine whether the event should be reported. If a system type event occurs, the Main Alerter immediately calls the configured system alerter as the fallback mechanism for unreportable errors.

Alerters

An *alerter* is an implementation class that can report an error. One alerter implementation class can have many configured instances that alert in different ways. Two alerter instances are preconfigured: the system alerter and the business process alerter.

The system alerter is the fallback mechanism that writes errors to the Alert service log file. The business process alerter is the default mechanism for executing a business process that sends notification. You can add more alerter instances through Application and more alerter classes by modifying the `alerter.properties` file.

Filters

A *filter* is an object that determines which events should or should not be reported. As with alerters, any number of filter instances can be configured per filter implementation class. Each filter instance can be associated with one or more alerters. An alerter can be configured with any number of filter instances. If more than one filter instance is assigned to an alerter, you are prompted to choose whether to combine the filters with AND or OR (default).

If the AND method is used, the alerter uses the minimum value returned by the set of filter instances to determine if an event should be reported. If an OR method is used, the alerter uses the maximum value returned by the filter set to determine whether it should report the event.

For example, consider an alerter instance with two filter instances defined. Each filter returns a return code with a particular weight. One filter instance returns ALERT (1) and the other returns DENY (0). If the alerter is configured in an AND method, the alerter uses the minimum value (DENY) and the event is not reported. If the OR method is specified, the alerter uses the maximum value (ALERT) and the event is reported.

One filter implementation is provided—the XPathFilter. XPathFilter instances enable you to specify an XPath query as criteria. If the Boolean result of evaluating the XPath query against the Event data is true, the XPathFilter returns ALERT. Otherwise, the XPathFilter returns DENY. For a discussion about how XPath expressions are evaluated for Boolean results, see www.w3.org/TR/xpath#section-Boolean-Functions.

Alert Service

The Alert service collects event data from Application and passes that data to the Main Alerter for event notification. It runs at a scheduled interval within the AlertProcess business process within Application. Event data is collected from the database (for information about which business processes have occurred) and from any active threads (for information about what is currently happening).

Alert Notification Business Processes

The Alert Notification business process sends e-mail to all members of the Notifications user group. This business process is a default method of alert notification. It can be changed to accommodate the alerting needs of the particular organization that is hosting Application.

Implementing the Alert Service

By default, when you implement and create a service configuration of the Alert service, the Alert service sends notification on any error or warning that happens in any business process, but this and other behavior is fully customizable. You can customize the Alert service using the following options:

- Modifying the predefined AlertNotification business process
- Configuring additional alerters and filter instances for the predefined classes
- Implementing new filter and alerter classes to plug in

To implement the Alert service, complete the following tasks:

1. Create an Alert service configuration. See *Managing Services and Adapters*.
2. Configure the Alert service in one of the following methods:
 - ◆ Accept the default behavior for the Alert service (to send e-mail notification on any business process error or warning).
 - ◆ Modify the predefined AlertNotification business process. For information, see *Modifying the Predefined AlertNotification Business Processes*.
 - ◆ Configure additional alerters and filter instances for the predefined classes. For information, see *Configuring Additional Alerters and Filter Instances for Predefined Classes*.
 - ◆ Implement new filter and alerter classes to plug in. For information, see *Implementing New Filter Classes to Plug In* and *Implementing New Alerter Classes to Plug In*.
3. Enable the Alert service schedule.
4. Use the Alert service in a business process.

Modifying the Predefined AlertNotification Business Processes

The AlertNotification business process performs notification. The default behavior for the AlertNotification process is to send an e-mail message to all members of the Notifications user group. You can change this behavior by modifying the AlertNotification business process using the Application Web administration tool.

From the Business Processes menu, select **Manager** and search for **AlertNotification**. Upload a new copy of the process and activate the new version.

If you choose to write your own business process, instead of using the AlertNotification business process, there are some extra steps that need to be taken in order for the system to use the business process defined in the alerter.properties file if it is something other than the default.

Edit the alerter.properties.in for the alerter.2.props.1.value to be set to the new custom business process, as shown in the following example from alerter.properties:

```
#
# BR Alerter
#
...
alerter.2.props.1.key=process
alerter.2.props.1.value=AlertNotification (this is the business process to run)
```

1. Stop Application.

2. Run `setupfiles.sh` (UNIX) or `setupfiles.cmd` (Windows).
3. Restart Application.
4. Run `setup_alerters.sh` (UNIX) or `setup_alerters.cmd` (Windows).

Note: The instance must be running during the last step.

Configuring Additional Alerters and Filter Instances for Predefined Classes

The alerter and filter configuration are defined in the `alerter.properties` file and loaded during the Main Alerter initialization process.

The `alerter.properties` file has the following general form:

```
# Alerters
alerter.#.id=<alerter_name>
alerter.#.class=<fully qualified class name>

# Alerter supported keys
# - multiple_filters_logic = [AND | OR]
#
# BP Alerter
# - process = <process_name>
#
# System Alerter
# - logger = <logger_name>

# alerter.#.props.#.key=<key name>
alerter.#.props.#.value=<value for key with corresponding number>

alerter.#.filter.#.name=<filter name>

# Filters
filter.#.class=<fully qualified class name>
filter.#.name=<unique filter name>
filter.#.criteria=<filter criteria data>
```

Edit the `alerter.properties` file when the Alert service is offline, or commit changes at run time through Application.

The `alerter.properties` file has the following default configurations:

Scheduled_AlertService – The `Scheduled_AlertService` process is the scheduled process that starts the main Alert process. The Alert process is the process that runs `AlertService`.

This business process has a daily default schedule. If you want to change the `Scheduled_AlertService` schedule, from the Deployment menu, select **Services > Schedules**.

AlertNotification – By default, the `AlertNotification` process processes reported events. The `AlertNotification` retrieves the e-mail addresses of the members of the Alert Notifications group and sends them e-mail with details of the event.

By default, there is one Alert Notifications group member, the Admin user. You can add or modify receivers of the notifications by adding or removing users from the Alert Notifications group.

The `alerter.properties` file has the following default properties:


```

#####
# Alerters settings #
#####

#
# System Alerter
#
# class implementations
alerter.1.class=com.sterlingcommerce.woodstock.alert.impl.SystemAlerter
# alerter's id (must be unique)
alerter.1.id=system_alerter

# key for first value of alerter properties
alerter.1.props.1.key=logger
# Value for first property of alerter properties
alerter.1.props.1.value=alerterlogger
alerter.1.props.2.key=multiple_filters_logic
# for this property the default value is "OR", the only other you can specify is "AND"
alerter.1.props.2.value=AND

# filter name
alerter.1.filter.1.name=xpathfilter_type_alerter

#
# BR Alerter
#
alerter.2.class=com.sterlingcommerce.woodstock.alert.impl.ProcessLauncherAlerter
alerter.2.id=bp_alerter

alerter.2.props.1.key=process
alerter.2.props.1.value=AlertNotification

alerter.2.props.2.key=multiple_filters_logic
alerter.2.props.2.value=OR

alerter.2.filter.1.name=xpathfilter_bp_status_notSuccess

#####
# Filters settings #
#####

#
# Filter for status other than success
#
# implementation class
filter.1.class=com.sterlingcommerce.woodstock.alert.filter.XPathFilter
# filter name (must be unique)
filter.1.name=xpathfilter_bp_status_notSuccess
# Filter's criteria (in this case we ignore BProcess of status "SUCCESS" or "WAITING"
and state "ACTIVE" )
filter.1.criteria=//EVENT/@type="BUSINESS_PROCESS" and not(//wf_status=0 or
//wf_status=3) and not(//wf_state=0)

#
# Filter for events type "ALERter"
#

```

```
filter.2.class=com.sterlingcommerce.woodstock.alert.filter.XPathFilter
filter.2.name=xpathfilter_type_alerter
filter.2.criteria=//EVENT/@type="ALERTER"
```

Implementing New Filter Classes to Plug In

A Filter implementation must implement the `com.sterlingcommerce.woodstock.alert.filter.Filter` interface.

For ease of development, the Alert service API provides a base filter implementation (`com.sterlingcommerce.woodstock.alert.filter.FilterBase`), which new filter implementations can extend. Derive the new implementation class from this base class and implement the `Decide` method. If you want your new filter to be loaded on initialization, define properties for it in the `alerter.properties` file or create a new instance of the filter in `Application`.

The filter `Decide` method of the object must return an integer value. The weight of the return code (the positive integer value) is used to determine if an event should be alerted. The greater the weight, the more important the event. A weight of zero indicates that the event should be ignored.

Two default weight constants are provided by the Filter interface: `ALERT` and `DENY`. The `ALERT` constant has a weight of one and the `DENY` constant has a weight of zero. These constants should be adequate for most filter implementations.

Implementing New Alerter Classes to Plug In

An Alerter implementation must implement the `com.sterlingcommerce.woodstock.alert.Alerter` interface.

For ease of development, the Alert Service API provides a base alerter implementation (`com.sterlingcommerce.woodstock.alert.impl.AlerterBase`), which new alerter implementations can extend. Extend the base class and provide the implementation method `report`. Then, add the Alerter into the `alerter.properties` file. If you do not add the alerter definitions to this file, your Alerter will not be used.

Archive Business Process Service

The Archive Business Process service searches for eligible records to archive from the Application live database tables.

Caution: This is an internal service that should not be used externally for steps in creating business processes because it is subject to change without notice, and use may cause unpredictable results and loss of data. This section is intended for information purposes only.

The following table provides an overview of the Archive Business Process service:

System name	Archive
Graphical Process Modeler (GPM) categories	All Services, System
Description	The service runs on a schedule. Copies records that have been flagged as eligible for archiving from the Live system database tables directly to disk.
Business usage	Eligible records are copied directly from the live tables to a file.
Usage example	Archiving business process information older than three days on a scheduled basis.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	Index Business Process service; Purge service; Purge Business Process Linkage service
Application requirements	No
Initiates business processes?	No
Invocation	Generally, this service runs by the Scheduler.
Business process context considerations	No
Returned status values	Messages are written to the Archive log.
Restrictions	Do not create multiple configurations of this service.
Persistence level	None
Testing considerations	None
Notes	If any step in the process fails, the service sends an e-mail notification.

How the Archive Business Process Service Works

The Archive function in Application is designed to be configured once, then run on a schedule. This service is configured as part of the Archive setup process.

When configuring this service, you specify what business process data should be archived, based on the number of days. You also choose how often to run the archive operation, based on your business needs or requirements. Once you have configured the Archive Business Process service, run the Archive business process to begin archiving data according to the parameters you have specified.

Attachment Parsing Service

The Attachment Parsing service extracts attachments from the MIME messages according to either the file name or file extension of the attachment. The following table provides an overview of the Attachment Parsing service:

System name	AttachmentParsingService
Graphical Process Modeler (GPM) categories	All Services, Translation
Description	The Attachment Parsing service retains the desired attachments in the MIME message of the primary document according to the filename or file extension, and removes unidentified attachments.
Business usage	In a business process, use this service to manipulate attachments in a primary document.
Usage example	After receiving a multipart MIME message that contains multiple documents, such as a Word attachment and an Excel attachment, you can use this service to forward one of the attachments. For example, if you want to send the Excel document to another department, use the Attachment Parsing service in a business process. The service deletes the Word document from the MIME message by passing in the specified file name or file extension. Then, using one of the communications adapters in the business process, such as the SMTP Send adapter, the Excel documents forwards to the other department.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	No, this is an internal service.
Business process context considerations	The configuration parameters and the documents are picked up by the adapter in the business process context.
Returned status values	<ul style="list-style-type: none">◆ Error occurred and the Message is xxx◆ Exit Status is x
Restrictions	The Attachment Parsing process operates only on the file name or file extension of the attachments. Use only the preconfigured service.
Persistence level	System default
Testing considerations	None

Business Scenario

In this scenario, the B2B Mail Client adapter is used to read a multipart mime message with two attachments. You want to filter the attachments and send the entire Excel document to a specific user. To do this:

1. Configure the B2B Mail Client adapter to extract all the mail headers into process data. On the adapter configuration URI page, you select a business process called SMTPExcelDoc to associate with this adapter configuration. The SMTPExcelDoc business process will process the e-mail read by B2B Mail Client adapter.
2. Configure two business processes: SMTPExcelDoc and Attachment_Parsing_Process.

SMTPExcelDoc Business Process Example

The following example illustrates the SMTPExcelDoc business process:

```
<process name="SMTPExcelDoc">
  <sequence name="xlsSequence">
    <assign to="fileExt">xls</assign>
    <assign to="xport-smtp-mailhost">00.000.00.000</assign>
    <assign to="xport-smtp-mailport">25</assign>
    <assign to="xport-smtp-mailto">John_Doe@company.com</assign>
    <assign to="xport-smtp-mailfrom" >John_Doe@company.com</assign>
    <spawn ref="ATTACHMENT_PARSING_PROCESS" />
  </sequence>
</process>
```

The SMTPExcelDoc business process spawns the Attachment_Parsing_Process business process. When a business process is instantiated as a spawn activity, as Attachment_Parsing_Process is in this example, a copy of process data is passed to the new business process.

Attachment_Parsing_Process Business Process Example

The following example illustrates the Attachment_Parsing_Process business process:

```
<process name="ATTACHMENT_PARSING_PROCESS">
  <sequence name="AnySequence">
    <operation name="trueOp">
      <participant name="AttachmentParsingService"/>
      <output message="Xout">
        <assign to="." from="*" />
        <assign to="fileExt" from="/ProcessData/fileExt/text()" />
      </output>
      <input message="Xin">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="SMTP Send">
      <participant name="SMTP_SEND_ADAPTER_SMTP_POP3_AUTO_TEST" />
      <output message="SMTP_SEND_ADAPTERInputMessage">
        <assign to="." from="*" />
        <assign to="xport-smtp-mailhost"
from="/ProcessData/xport-smtp-mailhost/text()" />
      </output>
    </operation>
  </sequence>
</process>
```

```

        <assign to="xport-smtp-mailport"
from="/ProcessData/xport-smtp-mailport/text() "/>
        <assign to="xport-smtp-mailto"
from="/ProcessData/xport-smtp-mailto/text() "/>
        <assign to="xport-smtp-mailfrom"
from="/ProcessData/xport-smtp-mailfrom/text() "/>
        </output>
        <input message="inmsg">
            <assign to="." from="*" />
        </input>
        </operation>
    </sequence>
</process>

```

Output from Business Process to Service

The following parameters can be passed to the service when it is run from a business process:

Parameter	Description
fileName	File name of the mime part which should be retained
fileExt	File extension of the mime parts those should be retained.

Implementing the Attachment Parsing Service

To implement the Attachment Parsing service, complete the following tasks:

1. Create an Attachment Parsing service configuration.
2. Use the Attachment Parsing service in a business process.

Auto Terminate Service

The Auto Terminate service terminates business processes that have been in a specified state for more than a specified number of days. The following table provides an overview of the Auto Terminate service:

System Name	AutoTerminateService
Graphical Process Modeler (GPM) categories)	All Services, System
Description	This service terminates business processes that have been in a pre-defined state for more than a specified length of time. By default, the Auto Terminate service terminates business processes that have been in a halted state for more than 14 days.
Business usage	This is a system service used to automatically terminate certain business processes.
Usage example	Used to automate the process of cleaning up business processes.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	Works within the Schedule_AutoTerminateService business process.
Application requirements	None
Initiates business processes?	No
Invocation	Invoked by the scheduler. By default, the Auto Terminate service runs once per day at 4:00 A.M.
Business process context considerations	None
Returned status values	None
Restrictions	Used with the bprecover.properties file.
Persistence level	System Default
Testing considerations	None

How the Auto Terminate Service Works

The Auto Terminate service is pre-configured and, by default, is scheduled to run each day at 4:00 A.M. The service checks for business processes that have been in a specified state for a specified length of time and then terminates them. By default, the Auto Terminate service checks for and terminates business processes that have been in a halted state for over 14 days. You can adjust these settings to suit your specific business needs. See *Overriding the bprecovery.properties File Settings* on page 26 for more information.

Implementing the Auto Terminate Service

A pre-configured instance of the Auto Terminate service is provided with Application and may be used in your business processes. You may also create new instances. To implement a new instance of the Auto Terminate service, complete the following tasks:


1. Create a new configuration of the Auto Terminate service. For basic information about creating service configurations, see *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Auto Terminate Service* on page 25.
2. Specify field settings for the service configuration in the Application Admin Console as necessary.

Configuring the Auto Terminate Service

You can specify settings for the Auto Terminate service in Application using the Admin Console, BPML, and by overriding settings in the `bprecovery.properties` file.

Creating a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the Auto Terminate service.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – Do not include the configuration in a service group at this time.◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.)◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. Note: For more information about service groups, see <i>Managing Services and Adapters</i> .
Run As User	Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list. Valid values: Any valid Application user ID.
Use 24 Hour Clock Display	Check the box to use the 24-hour clock instead of the default 12-hour clock.

Field	Description
Schedule	<p>Specify the scheduling information for running the Auto Terminate service.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If you select this field, the service does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service each day. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.

Setting Up the Service in the GPM

There is no configuration needed in the GPM for the Auto Terminate service.

Overriding the bprecovery.properties File Settings

The number of days a business process must be in a specified state before being terminated by the Auto Terminate service, and the specified state or states, are defined by properties in the bprecovery.properties file. The default settings are specified by the following lines:

```
auto_terminate_days=14
num_states=1
auto_terminate_state1=halted
auto_terminate_batch=1000
```

The default settings can be overridden using the customer_overrides.properties file. You can change the number of days before termination, change the specified state, or add additional states.

To change property file settings using the customer override property file:

1. In the *install_dir*/properties directory, locate (or create, if necessary) the customer_overrides.properties file.
2. Open the customer_overrides.properties file in a text editor.

3. Add the properties that you want to override, using the following format:

`bprecovery.PROPERTY_NAME=PROPERTY_VALUE`

Property	Description
<code>auto_terminate_days</code>	Number of days that a business process must be in one of the specified states before being terminated by the Auto Terminate service. Valid values: any positive integer. Default is 14.
<code>num_states</code>	Number of different specified business process states to use as a condition for terminating a business process. Must correspond to the number of auto_terminate_statex properties. For example, if there are 3 auto_terminate_statex properties, num_states must be set to 3. Default is 1.
<code>auto_terminate_statex</code>	<p>Business process state to use as a condition for terminating a business process. <i>x</i> = a digit corresponding to the number of the state. For example, the first state would be <code>auto_terminate_state1</code>.</p> <p>Default is <code>auto_terminate_state1=halted</code>.</p> <p>The number of auto_terminate_statex statements must correspond to the value of num_states. For example, if num_states is set to 3, there must be 3 auto_terminate_statex properties:</p> <ul style="list-style-type: none"> ◆ <code>auto_terminate_state1=1st state</code> ◆ <code>auto_terminate_state2=2nd state</code> ◆ <code>auto_terminate_state3=3rd state</code> <p>Valid states are:</p> <ul style="list-style-type: none"> ◆ <code>interrupted_man</code> ◆ <code>interrupted_auto</code> ◆ <code>halted</code> ◆ <code>waiting</code>
<code>auto_terminate_batch</code>	Maximum number of business processes that will be marked during each run of the <code>Schedule_AutoTerminateService</code> business process. Default is 1000.

For example, assume that you want to change the value of **auto_terminate_days** to 10 and add two more business process states, `interrupted_auto` and `interrupted_man`, in addition to the default state of `halted`.

To do so, add the following lines to the `customer_overrides.properties` file:

```

auto_terminate_days=10
bprecovery num_states=3
bprecovery auto_terminate_state2=interrupted_auto
bprecovery auto_terminate_state3=interrupted_man

```

4. Save and close the `customer_overrides.properties` file.
5. Stop and restart Application to use the new values.

The value of **auto_terminate_days** in the bprecovery.properties file can also be overridden using BPML in your business process using a statement in the following format:

```
<assign to="AUTO_TERM_DAYS" >new_value
```

The new value can be one of the following:

double – Doubles the value of **auto_terminate_days** in the bprecovery.properties file.

an integer value – A new integer value. Partial days can also be defined here. For example, 0.5 would specify one-half of a day.

Business Process Examples

The following BPML is the code used for the Schedule_AutoTerminateService business process included with Application. The Schedule_AutoTerminateService business process invokes the Auto Terminate service and provides e-mail notification of Auto Terminate activities. The e-mail notification provides details of the business processes that will be terminated. These details include the ID, data, and the time that the business process will be terminated. An e-mail notification is also generated after business processes are terminated.

```
<condition>count(//PrimaryDocument) > 0</condition>
</rule>

<rule name="Exist-PDOC-onfault">
  <condition>count(//PrimaryDocument) > 0</condition>
</rule>

<sequence>
  <operation name="SetLock">
    <participant name="SystemLockService"/>
    <output message="Xout">
      <assign to="LOCK_KEY">AutoTerminateHaltedBPS</assign>
      <assign to="DURATION">86400000</assign>
      <assign to="CLEAR_ON_START_UP">true</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="Service">
    <participant name="AutoTerminateService"/>
    <output message="Xout">
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
```

```

<operation name="unLock">
  <participant name="SystemLockService"/>
  <output message="Xout">
    <assign to="ACTION">unlock</assign>
    <assign to="LOCK_KEY">AutoTerminateHaltedBPS</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<sequence name="email">
  <choice name="exist pdoc">
    <select>
      <case ref="Exist-PDOC" activity="sending-email"/>
    </select>

    <sequence name="sending-email">
      <operation name="Get mailfrom">
        <participant name="UserService"/>
        <output message="UserServiceTypeInputMessage">
          <assign to="MethodName">getEmail</assign>
          <assign to="UserName">admin</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="mailfrom" from="//UserService/getEmail/text()"></assign>
        </input>
      </operation>

      <operation name="get mailto">
        <participant name="UserService"/>
        <output message="UserServiceTypeInputMessage">
          <assign to="MethodName">getEmail</assign>
          <assign to="UserName">admin</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="mailto" from="//UserService/getEmail/text()"></assign>
        </input>
      </operation>

      <operation name="SMTP Send">
        <participant name="SMTP_SEND_ADAPTER"/>
        <output message="SMTP_SEND_ADAPTERInputMessage">
          <assign to="xport-smtp-mailfrom" from="//mailfrom/text()"></assign>
          <assign to="xport-smtp-mailhost" from="'00.000.000.00'"></assign>
          <assign to="xport-smtp-mailport">25</assign>
          <assign to="xport-smtp-mailto" from="//mailto/text()"></assign>
          <assign to="xport-smtp-mailsubject">Automated Terminate Notification
        </assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
    </sequence>
  </choice>
</sequence>

```

```

        </input>
    </operation>

</sequence>
</choice>
</sequence>
<onFault>
    <sequence>
        <operation>
            <participant name="SystemLockService"/>
            <output message="Xout">
                <assign to="ACTION">unlock</assign>
                <assign to="LOCK_KEY">AutoTerminateHaltedBPS</assign>
                <assign to="." from="*"></assign>
            </output>
            <input message="Xin">
                <assign to="." from="*"></assign>
            </input>
        </operation>

<sequence name="email">
    <choice name="exist pdoc">
        <select>
            <case ref="Exist-PDOC-onfault" activity="sending-email-onfault"/>
        </select>

<sequence name="sending-email-onfault">
    <operation name="Get mailfrom">
        <participant name="UserService"/>
        <output message="UserServiceTypeInputMessage">
            <assign to="MethodName">getEmail</assign>
            <assign to="UserName">admin</assign>
            <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
            <assign to="mailfrom" from="//UserService/getEmail/text()"></assign>
        </input>
    </operation>

<operation name="get mailto">
    <participant name="UserService"/>
    <output message="UserServiceTypeInputMessage">
        <assign to="MethodName">getEmail</assign>
        <assign to="UserName">admin</assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
        <assign to="mailto" from="//UserService/getEmail/text()"></assign>
    </input>
</operation>

<operation name="SMTP Send">
    <participant name="SMTP_SEND_ADAPTER"/>
    <output message="SMTP_SEND_ADAPTERInputMessage">
        <assign to="xport-smtp-mailfrom" from="//mailfrom/text()"></assign>
        <assign to="xport-smtp-mailhost" from="'00.000.000.00'"></assign>
    </output>
    <input message="inmsg">
        <assign to="xport-smtp-mailhost" from="'00.000.000.00'"></assign>
    </input>
</operation>

```

```

        <assign to="xport-smtp-mailport">25</assign>
        <assign to="xport-smtp-mailto" from="//mailto/text()"></assign>
        <assign to="xport-smtp-mailsubject">Automated Terminate Notification
        </assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
        <assign to="." from="*"></assign>
    </input>
</operation>

    </sequence>
</choice>
</sequence>
</sequence>
</onFault>
</sequence>
</process>

```

The following BPML example doubles (by overriding) the number of Auto Terminate days defined in the `bprecovery.properties` file.

```

<operation name="Service">
    <participant name="AutoTerminateService"/>
    <output message="Xout">
        <assign to="AUTO_TERM_DAYS" >double</assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>

```

The following BPML example changes (by overriding) the number of Auto Terminate days to a half day.

```

<operation name="Service">
    <participant name="AutoTerminateService"/>
    <output message="Xout">
        <assign to="AUTO_TERM_DAYS" >0.5</assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>

```

B2B Lookup System Service

The B2B Lookup System service looks up trading profile information in the contract. The following table provides an overview of the B2B Lookup System service:

System name	B2B_LOOKUP_SYSTEM
Graphical Process Modeler (GPM) categories	All Services, System
Description	<p>Part of the system business process B2B_LOOKUP. This service always runs by the user services B2B_SEND, B2B_RESPOND, B2B_REQUEST_RESPONSE. The purpose of this service is to look up the trading profile information in the contract. This service takes the b2b-contract-id and b2b-message-mode that are in the business process context and searches the appropriate b2b-profile-id and sets that in the business process context. This service also looks for the appropriate transport business process for the transport associated with the contract.</p> <p>If the b2b-message-mode is:</p> <ul style="list-style-type: none">◆ Respond, the service uses the Production profile◆ Send, the service uses the Consumption profile◆ Request-response, the service uses the Production profile
Business usage	This service is not designed to be used as part of a business process. It is used by other services within Application.
Usage example	None
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	B2B Send, B2B Response, and B2B Request Response.
Application requirements	No
Initiates business processes?	No
Invocation	Run only by other services instead of a business process.
Business process context considerations	No

Returned status values	Name: B2B_LOOKUP Instance ID: 35 Service Name: B2B Lookup System service Contract ID=6fc193:e837daefd6:-7fc0 Profile ID=6fc193:e837daefd6:-7ff9 Transport Workflow Name=HTTP_SEND_ENVELOPE_OFF Transport Type=HTTP Message Mode=send
Restrictions	The trading profile information and contract must be configured before using this service. The service expects two parameters in the business process context: <ul style="list-style-type: none"> ◆ b2b-contract-id ◆ b2b-message-mode There should be only one instance of this service. It is a system service. It should not be used in any user-created business processes.
Persistence level	None
Testing considerations	This service is tested in conjunction with the following services: B2B Send, B2B Response, and B2B Request Response.
Notes	None

B2B Mail Client Adapter

The B2B Mail Client adapter enables Application to:

- Collect mail from an RFC 1725-compliant mail server using the POP3 protocol or an RFC 2060-compliant mail server using the IMAP protocol.

- Locate an appropriate business process or contract.

- Start a business process with the data appended.

For example, your company needs to collect EDI data from a remote mail server using the POP3 protocol. The B2B Mail Client adapter can collect the data from the appropriate mailbox and forward the results through a contract or business process to the translation services.

The following table provides an overview of the B2B Mail Client adapter:

System name	TEST_MAIL_CLIENT_ADAPTER
Graphical Process Modeler (GPM) categories	All Services, System
Description	Collects mail from an RFC 821-compliant mail server using the POP3 or IMAP protocol. The adapter can start a business process with the mail data it retrieves from the mail server. It can also place all the mail headers in the business process context for further processing.
Business usage	Polls the mail host at a specific interval to retrieve any mail in the mailbox. If there is no mail, the adapter tries again at the next interval.
Usage example	Some customers in remote locations send EDI data into a mail server mailbox through SMTP. A business process can be built to collect all new EDI data from the mailbox, and forward the results to the translation services.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms
Related services	No
Application requirements	Requires an RFC 821-compliant mail server and a mail server that supports POP3 or IMAP. The Mail server should be configured to mark mail as SEEN, ANSWERED, NEW, or DELETED when they are requested from B2B Mail Client adapter.
Initiates business processes?	Yes
Invocation	Not applicable (internal service)
Business process context considerations	This adapter can be configured to extract mail header information and place it into process data.

Returned status values	None
Restrictions	This adapter will connect to a mail server that supports POP3 or IMAP. The mail server should be RFC 821.
Persistence level	System default
Testing considerations	None

How the B2B Mail Client Adapter Works

In a business process that translates data, you can use a B2B Mail Client adapter configuration to start the business process.

1. The B2B Mail Client adapter detects data in a specified mailbox and initiates the appropriate business process or contract.
2. The Translation service translates the data that was collected by the B2B Mail Client adapter.

Process Data structure

Mail headers, message format, protocol, and contract ID information can be accessed in the process data of the business process document. They are written in the following format:

```
Mail headers
<ProcessData>
  <Mail_Client>
  <Headers>
  <{field name}>
  {---value---}
  </{field name}>
  </Headers>
  <Mail_Client>
</ProcessData>
```

Process Data Example

The following example shows how the process data might look:

```
<ProcessData>
<PrimaryDocument SCIObjectID="server:20187f:ef9c723ba3:-7fa4" />
<Mail_Client>
<Headers>
<From>"Boston, Test1" <test1@mycompany.com></From>
<Subject>xml file</Subject>
<To>"Boston, Test2" <test2@mycompany.com>, "Boston, Test3"
  <test2@mycompany.com></To>
<Date>Mon, 12 Aug 2002 10:48:28 +0800</Date>
<Received>by aaasgpxs00.boston.mycompany.com
  id <01C241AA.BCF67770@aaasgpxs00.boston.mycompany.com>; Mon, 12 Aug 2002 10:48:29
+0800</Received>
<MIME-Version>1.0</MIME-Version>
<Content-Type>application/octet-stream;
  name="smtpSetSubject.xml"</Content-Type>
<Content-Disposition>attachment;
```

```

    filename="smtpSetSubject.xml"</Content-Disposition>
<Message-ID><EA43E744516FD311A2FB00903B5C499CFF9B64@aaasgpxs00.boston.mycompany.com>
</Message-ID>
</Headers>
</Mail_Client>

</ProcessData>

```

Initiating a Business Process that Contains the B2B Mail Client Adapter

You can configure the B2B Mail Client adapter to look up either a predefined contract or a business process. When configuring the adapter, you must fill in Universal Resource Indicator (URI) information. The URI must be the same name that you entered in the User Name field. The URI name is then compared with the Userid (that matches a contract or business process). If a contract match is found, the business process ID is obtained from the contract and the adapter starts that business process. If a business process match is found, the adapter starts that business process.

Configuring Multiple URIs

You can configure multiple URIs when you configure the B2B Mail Client adapter for the first time. If you want to modify a URI, you need to change only the username and password of the URI. The same service configuration is used for multiple users. If you modify the service configuration, you need not change the business process. For example, if you want to bootstrap a business process configured in a second URI, you need to change the username and password for the second URI. The business process fails to bootstrap if the values of URI and User Name fields are different.

Business Process Example

The following example illustrates the B2B Mail Client adapter configuration called TEST_MAIL_CLIENT_ADAPTER being executed by a scheduler:

```

<process name="Schedule_TEST_MAIL_CLIENT_ADAPTER">
  <sequence>
    <operation name="Service">
      <participant name="TEST_MAIL_CLIENT_ADAPTER"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Implementing the B2B Mail Client Adapter

To implement the B2B Mail Client adapter, complete the following tasks:

1. Create a B2B Mail Client adapter configuration. For information, see *Managing Services and Adapters*.

2. Configure the B2B Mail Client adapter. For information, see *Configuring the B2B Mail Client Adapter* on page 37.

Note: When configuring the B2B Mail Client adapter, consider that you cannot pass parameters to the B2B Mail Client adapter from BPML or configure the adapter using the GPM.

3. Use the B2B Mail Client adapter in a business process. For information, see *Initiating a Business Process that Contains the B2B Mail Client Adapter* on page 36.

Configuring the B2B Mail Client Adapter

To configure the B2B Mail Client adapter, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for this adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list.
Default Message Access Protocol	Select the mail accessing protocol: POP3 or IMAP. Required.
Default Mail Server	Name of the mail server that Application is to connect to. This is any valid DNS name or a dotted decimal address.
Default Mail Server Port	Port number that is monitored for activity by the remote system. This must be a valid port and must be 1 - 5 numeric characters.
User Name	Name that identifies the mailbox to provide when connecting to a mail server. This field is case-sensitive.
Password	Password used for authentication when connecting to a mail server. This field is case-sensitive.
Connection Retries	The number of time Mail Client adapter tries to connect to the mail server when there is a connection error. Optional. Minimum valid value is 1 (default is 3).
Retry Interval (Sec)	The retry interval specifies in seconds. Optional. Minimum valid value is 1 (also the default).
Max Messages per Session	Maximum number of messages that can be processed per mail download session. Optional. Minimum valid value is 1 (default is 500).

Field	Description
Remove inbox mail messages	To remove the mail message from the inbox after successfully downloading them. Only configurable if using IMAP protocol. If POP3 is used to access the mailbox, the messages will be removed from the inbox after they have been successfully downloaded. Optional. Valid values are Yes and No.
SMIME Encryption User Certificate	Not used in this release. Leave at default.
SMIME Decryption User Certificate (System Store)	Not used in this release. Leave at default.
SSL	Whether to use SSL to connect to mail server. Optional. Valid values are Must and None. If Must is selected, you must configure the SSL parameters that follow Key Certificate Passphrase.
Key Certificate Passphrase	The passphrase for the key certificate. Optional.
Cipher Strength	The cipher strength use for SSL. Optional. Valid values are Strong, Weak, and All.
Key Certificate (System Store)	Object ID that contains the privatekey and public certificate. The trading partner trusted certificate authority signs the Certificate. Optional.
CA Certificates	Object ID that contains the trusted certificate authority public certificate. Optional.
Add New URI	Click add to add a new Uniform Resource Indicator (URI).
Note: The following four fields are displayed only if Add New URI is selected:	
URI	Uniform Resource Indicator. Use the same name that is provided in the User Name field above. This field is case-sensitive and is 1 - 8 alphanumeric characters beginning with a letter.
Business Process	Name of the business process you want the B2B Mail Client adapter to start. This field is valid only if no contract is defined. Either a contract or a business process can be associated with this adapter, but not both.
Contract	Name of the contract to associate with this adapter. This is valid if no business process is defined. Either a contract or a business process can be associated with this adapter, but not both.
Send Raw Messages	Whether to send raw messages that preserve the mail headers in the body of the message. Valid values are Yes and No.
Insert into Process Data	Whether to extract all the mail headers information from the mail message and put them into process data. Required. Valid values are All mail header fields and Selected mail header fields (default is All mail header fields). If Selected mail header fields is chosen, Application will only attempt to extract the values you choose from the following four pages.
Mail Header: Originator Fields	Only displayed if Selected mail header fields was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: From, Resent-from, Sender, Resent-Sender, Reply-to, Resent-reply-to. Optional.

Field	Description
Mail Header: Receiver Fields	Only displayed if Selected mail header fields was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: To, Resent-to, CC, Resent-cc, BCC, Resent-bcc. Optional.
Mail Header: Reference Fields	Only displayed if Selected mail header fields was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: Message-id, Resent-message-id, In-Reply-to, References, Keywords. Optional.
Mail Header: Other Fields	Only displayed if Selected mail header fields was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: Subject, Comments, Encrypted, Date, Resent-date, Attachments. Optional.
Run as User	Enter (or select from the list) the user ID to be associated with business process instances of this service.
Use 24 Hour Clock Display	Select to specify times for this schedule using the 24 hour clock. Leave blank to use 12 hour clock and AM and PM.
Schedule	<p>Information about scheduling the adapter configuration to run and to start the specified business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, this service does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the month Valid values are the day of the month (including the last day of the month (LDM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup.

B2B Request/Response System Service

The B2B Request/Response System service requests messages from and responds to messages from trading partners using trading profile information in the contract. The following table provides an overview of the B2B Request/Response System service:

System name	B2B_REQUEST_RESPONSE_SERVICE
Graphical Process Modeler (GPM) category	All Services
Description	Requests a response to messages from trading partners using trading profile information in the contract. A contract has two trading profiles: a production profile and a consumption profile. For example, there is a contract with trading partner A as production profile and trading partner B as consumption profile. Including this service in a business process will send a message to trading partner A.
Business usage	Use this service in a business process when you want to send a message to the production profile of the contract.
Usage example	This service can be used to respond to messages from trading partners referenced within the production profile of the contract.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	B2B Send service, B2B Respond service, B2B Lookup service
Application requirements	Uses trading profile information; in particular, it uses the parameters specified in the contract production profile transport to send messages.
Initiates business processes?	It is statically preconfigured to start the B2B Lookup business process.
Invocation	This service may not be the first in the business process. It may be just before the end of the participant list.
Business process context considerations	No
Returned status values	InvokeBusinessProcess(Spawn) successfully invoked instance xxx of B2B_LOOKUP:1
Restrictions	The trading profile information and contract must be configured before using this service. There is no limit to the number of configurations, but it is strongly suggested that you use the preconfigured service.
Persistence level	None
Testing considerations	None
Notes	None

B2B Send System Service

The B2B Send system service sends messages to trading partners. It does this by retrieving and using information from trading profiles and the associated trading profile contract to identify how and where to send information. After creating a business process that uses the B2B Send service, the same business process can be associated with any number of contracts to send messages to the consumption profile of the contract. The following table provides an overview of the B2B Send system service:

System name	B2B_SEND
Graphical Process Modeler (GPM) category	All Services
Description	The B2B Send system service sends messages to trading partners. It does this by retrieving and using information from trading profiles and the associated trading profile contract to identify how and where to send information.
Business usage	Use this service in a business process to send a document or message to a trading partner identified as the consumption profile (receiving partner) in a contract.
Usage example	Send messages to a trading partner.
Preconfigured?	Yes
Requires third-party files?	Yes
Platform availability	All supported Gentran Integration Suite platforms
Related services	<p>This service is one of four services that are used in Gentran Integration Suite for B2B communications using trading profiles. The other three services are:</p> <ul style="list-style-type: none">◆ B2B Respond service – Responds to messages from a production (sending) trading partner using trading profile information in the contract.◆ B2B Request Response service – Uses the production (sending) profile of a contract to send a request to and retrieve a response from the production trading partner.◆ B2B Lookup service – Finds trading partner contract records and retrieves profile information. It is a system service included automatically in your business process if you include the B2B Respond service, the B2B Request Response service, or the B2B Send service. <p>Two other services provide EDI documents that can be sent using the B2B Send adapter. These services are:</p> <ul style="list-style-type: none">◆ EDI Encoder service – Determines which enveloping services need to run and sends the document to the translator for translation, if needed.◆ EDI Enveloping service – Determines which business processes need to run to apply EDI envelopes. <p>Note: The EDI Encoder and Enveloping services prepare the document to be sent.</p>

Application requirements	This service runs two subprocesses: the B2B_LOOKUP business process and the transport business process specified in the consumption profile for the Send message mode. The subprocesses use information from a trading profile, including the consumption profile transport configuration parameters from a contract, to send a message.
Initiates business processes?	Pre-configured to start the B2B_LOOKUP business process, which launches the transport workflow identified in the profile's transport settings.
Invocation	Runs as part of a business process. It cannot be the first participant in a business process unless the business process runs as a subprocess, because it requires the <i>b2b-contract-id</i> as an input parameter.
Business process context considerations	None
Returned status values	InvokeBusinessProcess(Spawn) successfully invoked instance xxx of B2B_LOOKUP:1.
Restrictions	The trading profile information and contract must be configured before using this service.
Persistence level	None
Testing considerations	If there is an error, go to the business process monitor (select Business Process > Monitor > Current Processes) and check the status. Also navigate through the subprocesses under this service in the process monitor.

About B2B Services and Trading Partner Setup

B2B services are dynamic services, independent of transport protocols or adapters, trading profiles, and contracts, that determine how to exchange data between trading partners. By using these services in a business process, you can eliminate the need to configure in the business process many of the details necessary to define this data exchange. All of this information has been defined in the trading profiles, and the B2B services dynamically look up the requirements in the profiles.

To communicate with trading partners, you must define a trading partner record and use at least one of the following B2B services in every business process (unless you are using the EDIINT Message service):

- B2B Lookup Service
- B2B Send Service
- B2B Respond Service
- B2B Request-Response Service

How the B2B Send Service Works

The B2B Send service is preconfigured and installed as part of Gentran Integration Suite. No additional configuration of this service is necessary.

When using the B2B Send service in a business process:

1. The service runs as part of the business process.
2. This service starts the B2B Lookup service.
3. The B2B Lookup service locates:

- ◆ The consumption profile of the contract
- ◆ The transport protocol of the consumption profile
- ◆ The appropriate transport business process for the profile transport protocol (default), or the business process associated with the send message mode in the consumption profile, if specified

The default transport business processes are:

HTTP_SEND_ENVELOPE_OFF

FTP_SEND_ENVELOPE_OFF

SMTP_SEND_ENVELOPE_OFF

4. The B2B Lookup service either runs the default business process, if none was specified in the profile, or runs the business process specified in the consumption (receiver) profile for Send message mode.
5. The B2B Lookup service starts the appropriate communications adapter.
6. The communications adapter sends a message to the receiver transport server that is identified in the transport section of the profile information.

The B2B Send service always runs a subprocess. (It is configured to start the B2B Lookup business process, but this process can be changed in the service configuration.) This process in turn runs another sub-process that sends the message. This is either one of the default transport business processes specified above or another business process identified in the consumption profile.

Note: After a business process is created that uses the B2B Send service, the same business process can be associated with any number of contracts to send messages to the consumption profile of the contract.

Business Scenario

You have invoices that are currently in XML format that you need to translate and send to several business partners. These partners require their information be sent in a variety of ways: HTTP, FTP, AS2, and SMTP, as well as in a variety of formats, including EDI 810. You want a single business process that can process all invoices, translate them to the correct format, create the necessary document envelopes, and send the invoices to each customer using the correct transport method.

To solve the business scenario, use the following example that demonstrates sending an EDI 810 document using HTTP:

1. Create a business process (SendOutBoundDocuments) that uses the B2B Send service.
2. Configure a consumption trading profile, a production trading profile, and a contract for these two profiles.

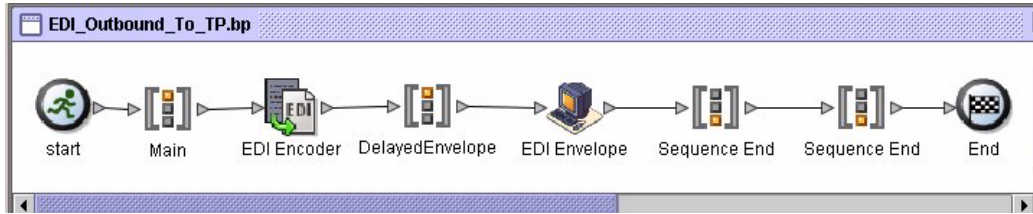
The trading partner information should include a map to translate the XML document to an EDI 810 document as well as all of the required envelope and transport information. It also should reference the SendOutBoundDocuments business process in the contract definition.

3. Create a business process (EDI_Outbound_To_TP) that uses the EDI Encoder service and the EDI Enveloping service.

This business process uses the EDI Encoder and EDI Enveloping services to translate this information, create the document envelopes, and send it to the trading partner using the B2B Send service.

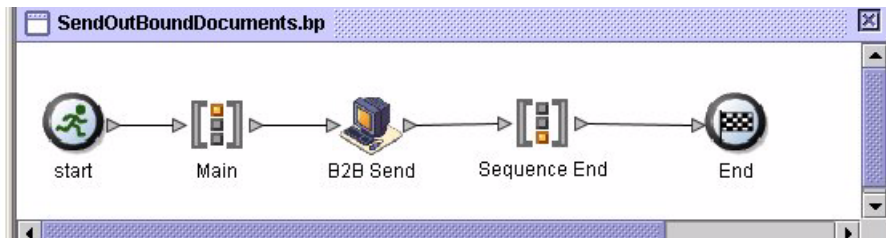
This business solution is described for both the GPM and for the Business Process Modeling Language (BPML).

GPM Configuration for the EDI_Outbound_To_TP Business Process Example



1. The EDI_Outbound_To_TP business process receives the XML file as the primary document.
2. The EDI Encoder service takes the sender, receiver, and lookup alias as input parameters. Based on this information, it determines which enveloping services to run and sends the document to the translator, if required.
3. The EDI Enveloping service envelops the document using the parameters specified in the Trading Partner configuration and then runs the SendOutBoundDocuments process specified in the contract.
4. The SendOutBoundDocuments business process receives the enveloped document as the primary document.
5. The B2B Send Service uses the b2b-contract-id that is in process data to identify from the trading partner setup how and where to send the document. In this example, the document is sent using HTTP.

GPM Configuration for the SendOutBoundDocuments Business Process Example



The SendOutBoundDocuments Business Process consists only of the B2B Send service. In this case, it always runs as a subprocess and the b2b-contract-id is in process data when it is run. There is no additional configuration required for this service in the GPM.

Process Data Example

The example below shows that the process data contains the b2b-contract-id that was output to process data from the EDI Enveloping service. This information is required in the business process for the B2B Send service to work.

```

<ProcessData>
  <ENVELOPE_TYPE>ST SE</ENVELOPE_TYPE>
  <ReceiverIdentityID>f96c:fb0eba2655:-79ed</ReceiverIdentityID>
  <SenderIdentityID>f96c:fb0eba2655:-7a26</SenderIdentityID>
  <PrimaryDocument SCIObjectID="L2000-000248:f96c:fb1bf8051e:-6019"/>
  <TransactionSetIDCode>820</TransactionSetIDCode>
  <GroupVersionReleaseIDCode>004010</GroupVersionReleaseIDCode>
  <GroupFunctionalIDCode>RA</GroupFunctionalIDCode>
  <InterchangeSenderID>MYID</InterchangeSenderID>
  <InterchangeReceiverID>MYTPsID</InterchangeReceiverID>
  <InterchangeControlNumber>000000002</InterchangeControlNumber>
  <InterchangeTestIndicator>I</InterchangeTestIndicator>
  <CONTRACT_FOUND>YES</CONTRACT_FOUND>
  <WFD_NAME>SendOutBoundDocuments</WFD_NAME>
  <b2b-contract-id>f96c:fb0eba2655:-7848</b2b-contract-id>
</ProcessData>

```

The SendOutBoundDocuments business process is found in the contract as the process to invoke.

B2b-contract-id in process data is available for the B2B Send adapter.

Contract Configuration with the SendOutBoundDocuments Business Process Example

The SendOutBoundDocuments business process is specified in the contract as shown in the Trading Partner-Contracts Business Process setup.

Business Process Modeling Language (BPML) Example

For a generic business process, the values have to be dynamically generated, based on the consumption profile. The following example shows the corresponding business process solution using BPML and the (hard-coded) AcceptorLookupAlias, ReceiverID, and SenderID:

EDI_Outbound_To_TP Business Process

```

<process name="EDI_Outbound_To_TP">
  <sequence name="Main">
    <operation name="EDI Encoder">
      <participant name="EDIEncoder"/>
      <output message="EDIEncoderTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="AcceptorLookupAlias">820</assign>
        <assign to="ReceiverID">MYTPsID</assign>
        <assign to="SenderID">MYID</assign>
      </output>
      <input message="inmsg">
      </input>
    </operation>
    <sequence name="DelayedEnvelope">
      <operation name="EDI Envelope">
        <participant name="EDIEnvelope"/>
        <output message="EDIEnvelopeTypeInputMessage">
          <assign to="." from="*"></assign>
          <assign to="MODE">DEFERRED</assign>
          <assign to="RECEIVER_ID">MYTPsID</assign>
          <assign to="SENDER_ID">MYID</assign>
          <assign to="AcceptorLookupAlias">820</assign>
        </output>
        <input message="inmsg">
        </input>
      </operation>
    </sequence>
  </sequence>
</process>

```

The AcceptorLookupAlias, ReceiverID, and SenderID are assigned in the EDI Encoder.

SendOutBoundDocuments Business Process

```

<process name="SendOutBoundDocuments">
  <sequence name="Main">
    <operation name="B2B Send">
      <participant name="B2B_SEND"/>
      <output message="B2B_SEND_SERVICEInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

The Business Process Monitor shows the order in which the business processes ran.

Status	ID	Name	State	Started	Ended	Expires	Parent/Child
	43518	HTTP_SEND_ENVELOPE_OFF	Active	3/11/04 1:30:01 PM			▲
	43517	B2B_LOOKUP	Waiting	3/11/04 1:30:01 PM			▲ ▼
	43516	SendOutBoundDocuments	Waiting	3/11/04 1:30:01 PM			▲ ▼
	43515	X12EnvelopeUnified	Completed	3/11/04 1:29:57 PM	3/11/04 1:30:01 PM		▲ ▼
	43514	EDI_Outbound_ToTP.bp	Completed	3/11/04 1:29:56 PM	3/11/04 1:29:56 PM		▼

Implementing the B2B Send Service

The B2B Send service is preconfigured and installed as part of Gentran Integration Suite. No additional configuration of this service is necessary. However, to use the B2B Send service, you must create and enable a business process that includes the preconfigured B2B Send service.

To implement the B2B Send Service, complete the following tasks:

1. Configure the trading profile and a corresponding contract and obtain the b2b-contract-id from the trading profile contract.
2. Use the B2B Send service in a business process.

Editing the B2B Send Service

Should you need to edit configuration setup for the B2B Send service, you can edit the B2B Send service using the GPM. The following example shows the GPM parameters for the B2B Send service:

Name	Value	Use XPATH?
b2b-message-mode	send	<input type="checkbox"/>
COPY_SERVICE_PARMS	TRUE	<input type="checkbox"/>
defaultWFID		<input type="checkbox"/>
envelopingOn		<input type="checkbox"/>
INVOKE_MODE	SYNC	<input type="checkbox"/>
PARM_LIST		<input type="checkbox"/>
SPAWN	FALSE	<input type="checkbox"/>
tradingPartnerMode	USE_CONSUMPTION_PROFILE	<input type="checkbox"/>
USE_STATIC_WFD_NAME	TRUE	<input type="checkbox"/>
useDefaultWFID	TRUE	<input type="checkbox"/>
userWFID		<input type="checkbox"/>
WFD_NAME	B2B_LOOKUP	<input type="checkbox"/>
WFD_VERSION		<input type="checkbox"/>
xport-b2b-mode	on	<input type="checkbox"/>

Note: The dimmed values were generated from the preconfigured B2B Send system service configuration.

The following table describes the fields used as input to the B2B Send service in the business process:

Parameter Name	Description
Config	Name of the Service configuration. Required.
b2b-contract-id	The contract ID with the transport information to be used to send the message. This value is input from the business process in process data. Valid value is any valid contract ID.

Parameter Name	Description
b2b-raw-response	Whether to receive a raw response. Optional. Valid values are: <ul style="list-style-type: none"> ◆ true - receive raw response ◆ false - do not receive raw response (Default) Can be used interchangeably with xport-raw-response.
xport-raw-response	Whether to receive a raw response. Optional. Valid values are: <ul style="list-style-type: none"> ◆ true - receive raw response ◆ false - do not receive raw response (Default) Can be used interchangeably with b2b-raw-response.

The following table describes the output parameter from the subworkflows run by the B2B Send service.

Parameter Name	Description
b2b-profile-id	The consumption profile ID from the contract.

Usage Example

The following examples of the GPM and BPML describe the most basic implementation of the B2B Send service in a business process. In this case, the b2b-process-id is known and directly input to the service. The Assign activity is used to identify the b2b-contract-id in the business process.

GPM Example

In the business process, precede the B2B Send service with the Assign Activity and configure it to assign the contract ID to b2b-contract-id or the contract name to b2b-contract-name.



The following figure shows the graphical view of the GPM parameters for the Assign activity. The contract ID (3c9217:f5894f9-24ca) is assigned to b2b-contract-id. This is a required parameter for the B2B Send service and is found in the trading partner contract information. The B2B Lookup service looks up the

trading partner transport information and uses the specified transport protocol (such as HTTP, SMTP, or FTP) to send the message.

Property Editor - Assign	
Name	Value
append	false
constant	3c9217:f5894f94f9:-24ca
from	
to	b2b-contract-id

BPML Example

The following example provides BPML that corresponds the previous GPM example:

```
<process name="ExampleB2BSetup">
  <sequence name="Main">
    <assign name="Assign" to="b2b-contract-id">3c9217:f5894f94f9:-24ca</assign>
    <operation name="B2B Send">
      <participant name="B2B_SEND"/>
      <output message="B2B_SEND_SERVICEInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

The following example provides the process data in the business process when the B2B Send service runs, including the b2b-contract-id that must be in process data for the service to complete successfully:

```
<ProcessData>
  <PrimaryDocument SCIOBJECTID="L2000-000248:f96c:fb0bd69aa1:-73af"/>
  <b2b-contract-id>3c9217:f5894f94f9:-24ca</b2b-contract-id>
</ProcessData>
```

Example Using b2b-raw-response

```
<operation name="B2B_SEND">
  <participant name="B2B_SEND"/>
  <output message="B2B_SEND_ADAPTERInputMessage">
    <assign to="b2b-contract-id">xxx</assign>
    <assign to="b2b-raw-response">true</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

B2B SMTP Client Adapter

The B2B SMTP Client adapter uses SMTP to send documents to remote and local mail servers. The following table provides an overview of the B2B SMTP Client adapter:

System name	B2B_SMTP_CLIENT_ADAPTER
Graphical Process Modeler (GPM) categories	All Services, System
Description	This adapter will not be referenced in a business process. The B2B Lookup service references this adapter by the name B2B_SMTP_CLIENT_ADAPTER.
Business usage	This adapter is used as a communications adapter only by the B2B Lookup Service.
Usage example	None
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	B2B Lookup service
Application requirements	An internal SMTP server to send mail.
Initiates business processes?	No
Invocation	This service runs only by the B2B Lookup service.
Business process context considerations	None
Returned status values	None
Restrictions	This service is for use only by Application services.
Persistence level	System default
Testing considerations	None

Backup Business Process Service

The Backup Business Process service backs up (archives) business process data to physical media prior to removing it from the live system. The following table provides an overview of the Backup Business Process service:

System Name	BackupService
Graphical Process Modeler (GPM) categories	None
Description	Backs up business process data to physical media after the business process completes or terminates and is then indexed. Data removal is separated from data archiving. Data removal is done by the Purge service after the business process life span (as specified in the Archive Manager) is exceeded.
Business usage	This is a internal only service that runs based on settings specified in the Archive Manager.
Usage example	This service should not be included in a business process; it is used internally for archive processing.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	Restore Business Process service, Index Business Process service
Application requirements	Adequate offline storage space must be available to the service for writing the back up data files.
Initiates business processes?	No
Invocation	Invoked by the system business process for archiving, BackupService.bp
Business process context considerations	None
Returned status values	None
Restrictions	There must be only one configuration of the Backup Business Process service.
Persistence level	System Default
Testing considerations	None

How the Backup Business Process Service Works

The Backup Business Process service is invoked by the BackupService business process based on settings in the Archive Configuration section of the Archive Manager. By default, the Backup Business Process

service will run daily at 2:00 a.m. Business processes specified in the Archive Manager to be archived will be saved in a backup set. The Backup Business Process service backs up the specified business process data and removes it from the database.

Caution: To prevent data loss, do not delete a backup set, even if there is an error, without first copying the backup set to some form of persistent storage.

Beginning with version 4.1.1, Application supports incremental backups. The Backup Business Process service divides all business processes that are eligible for archiving into chunks and processes each chunk independently. If an error occurs while processing a chunk, the chunk that had the error is invalidated. This eliminates the need to rerun the entire backup set because of an error in that chunk.

Incremental backup also allows you more control over the backup process. Each time you run the Backup Business Process service, you can specify the maximum number of business processes per chunk, the maximum number of chunks to process and the maximum number of days per backup set. For example, if you only want to back up a single business process, you can set the maximum number of business processes per chunk to 1 and the maximum number of chunks to 1.

Backup sets are stored in a directory with a name based on the date and time stamp when the backup set was created. For example, a backup set created on May 3, 2006 at about 4:22 p.m. might be named 20060503_162234. Each chunk of the backup set has a separate data directory named using the following naming convention *DATAchunkNumber*. For example, the first chunk is stored in DATA00000 and the tenth chunk is stored in DATA00009. Only good data directories are kept in backup sets.

If an error occurs while backing up a chunk of business processes, that chunk is not backed up in that set and will be returned for the next backup. The system deletes the directory on disk that holds the chunk data that was written before the error, but the processes in the chunk are not deleted. If the error is later resolved, the chunk will then be backed up. If not, a notification is sent to the system administrator.

The most likely errors while backing up chunks are not due to bad data in business processes. Most errors are due to a transient condition, like a timeout from the overloading of the database. When the transient condition is resolved and the archive is run again, the error does not occur, and the chunk is backed up normally. For errors that do not go away, contact customer support to determine a course of action.

Table data is written to files using Java serialization. The files are named using the name of the table being backed up according to the following convention: *tableName.dat*. For example, the backup file for the WORKFLOW_CONTEXT table is named workflow_context.dat. Any documents on disk that are associated with table data are copied to the backup set.

Note: Backing up documents on disk might add significantly to your total backup time when, for example, you copy large files to a remotely mounted file system or to a multiple drive file system.


Implementing the Backup Business Process Service

The Backup Business Process service is preconfigured as BackupService. Do not create additional configurations. The BackupService configuration can be edited, if needed, in the Application Admin Console. For basic information about editing service configurations, see *Managing Services and Adapters*.

Configuring the Backup Business Process Service

You can specify field settings in Application, using the Admin Console.

Field	Description
Name	Unique and meaningful name for the service configuration. Required. Do not change the preconfigured name.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – Do not include the configuration in a service group at this time.◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.)◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. Note: For more information about service groups, see <i>Managing Services and Adapters</i> .
Thread Pool Size	Number of execution threads allocated to archiving. Required. Default is 8. Note: Although allocating more threads usually increases performance, never allocate more threads than the number of tables plus one. Allocating too many threads can decrease performance.
Number of Business Processes per Backup Set Chunk	Maximum number of business processes grouped together within a single chunk of a backup set. Required. Default is 10,000.
Maximum Number of Chunks per Invocation	Maximum number of chunks that will be backed up in a single backup set. Required. Default is 25.
Compress Data	Whether to compress the data before archiving. Valid values: <ul style="list-style-type: none">◆ true – Compress the data. This reduces the amount of disk space required for storing the backup set.◆ false – Do not compress the data. This improves backup speed and performance at the expense of disk space.
Maximum Backup File Size	Maximum size of a backup file before creating a new file. Required. Default is 1.
Archive File Size Unit of Measure	Unit of measure used for Maximum Backup File Size . Required. Valid values: <ul style="list-style-type: none">◆ K – Kilobyte◆ M – Megabyte◆ G – (Default) Gigabyte◆ T – Terabyte
Maximum Number of Days per Backup Set	Maximum number of days that will be backed up in one backup set. Required. Default is 2.

Field	Description
Run as User	<p>Applies to the scheduling of the business process.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid value is any valid Application user ID.</p> <p>Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>
Use 24 Hour Clock Display	<p>If selected, the service will use the 24-hour clock instead of the default 12-hour clock.</p>
Schedule	<p>Information about scheduling the business process invoked by the Backup Business Process service.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, the service does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.

Batch Processor Service

The following table provides an overview of the Batch Processor service:

System name	BatchProcessorService
Graphical Process Modeler (GPM) category	All Services
Description	The Batch Processor service is used to merge XML documents or XML document fragments. If specified, each XML document can be translated through a map before being merged.
Business usage	If you want to create one XML document out of multiple XML documents, but only fragments of the documents need to be merged and some values need to be replaced, use the Batch Processor service.
Usage example	Example with UCCnet Edition: The UCCnet Edition uses the Batch Processor service when sending batches to UCCnet. Fragments of RCI documents need to be combined into one document, and some values need to be replaced. The Batch Processor sends each RCI document through a map, which replaces some of the values in the document. Then a fragment of each document is parsed out and added to an output document. Each RCI document goes through the same process. Once all documents have been processed and added to the output document, the output document is placed as the PrimaryDocument.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – The service successfully created an output document◆ Error – The service was not successful in creating an output document.
Restrictions	<ul style="list-style-type: none">◆ Only XML documents to be used◆ Any number of configurations can be created, but typically a single configuration can be used by any number of business processes.
Persistence level	None

Testing considerations To test it, create a configuration or use the existing configuration, then run it from a business process as shown in the business process examples later in this section.

Implementing the Batch Processor Service

To implement the Batch Processor service, complete the following tasks:

1. Create a Batch Processor service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Batch Processor service. For information, see *Configuring the Batch Processor Service*.
3. Use the Batch Processor service in a business process.

Configuring the Batch Processor Service

To configure the Batch Processor service, you must specify field settings in Application and in the GPM.

Field	Description
Config	Name of the service configuration.
prefix_document (prefix_document)	The content of the XML document referenced will be added to the beginning of the output document. If an XML document does not exist with the value specified, XPath will be applied to the value and the text retrieved by xpath added to the beginning of the document (xpath and text needs to be placed in ProcessData). If the xpath does not exist either, nothing will be added. The value has to match the name of an XML document or xpath. If neither exists, nothing will be added to the beginning of the output document. Optional.
suffix_document (suffix_document)	The content of the XML document referenced will be added to the end of the output document. If an XML document does not exist with the value specified, xpath will be applied to the value and the text retrieved by xpath added to the end of the document (xpath and text needs to be placed in ProcessData). If the xpath does not exist either, nothing will be added. The value has to match the name of an XML document or xpath. If neither exists, nothing will be added to the beginning of the output document. Optional.
input_document (input_document)	This XML document contains references to other XML documents that will be added between the prefix_document and suffix_document. Each of the XML documents referenced might be partially added, translated using a map, and/or have values replaced. The value has to match the name of an XML document. Optional.

Business Process (BPML) Example

The following BPML runs the Batch Processor service. It has three wfd parameters defined: prefix_document, suffix_document, and input_document:

```
<operation>
  <participant name="BatchProcessorService"/>
  <output message="outmsg">
    <assign to="." from="*" />
    <assign to="prefix_document">PrefixDocument</assign>
    <assign to="suffix_document">SuffixDocument</assign>
    <assign to="input_document">InputDocument</assign>
  </output>
</operation>
```

```

    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>

```

In this example, consider that:

The `prefix_document` name is `PrefixDocument` and `suffix_document` name is `SuffixDocument`. If the `PrefixDocument` exists, content of this XML document will be added to the beginning of the output document.

If the `PrefixDocument` does not exist, the Batch Processor Service will look in `ProcessData` and apply `xpath` to `PrefixDocument`. If `/ProcessData/PrefixDocument` element exists, the value for that element will be added to the beginning of the output document.

If the `PrefixDocument` does not exist, nor does `/ProcessData/PrefixDocument` element, nothing will be added to the beginning of the output document.

The same processing rules that apply to `PrefixDocument` also apply to `SuffixDocument`, except that the `SuffixDocument` will be added to the end of the output document

The `input_document` name is `InputDocument`. If `InputDocument` does not exist, an error will be thrown. The `InputDocument` contains references to other XML documents that enable the service to pass the output to the `PrimaryDocument`.

The following example shows a `PrefixDocument` that is added to the beginning of the XML output document:

```
<BatchProcessorOutput>
```

The following example shows a `SuffixDocument` that is added to the end of the XML output document:

```
</BatchProcessorOutput>
```

The following example shows an `InputDocument`:

```

<batchProcessor>
  <fragment docname="Doc1" mapname="BatchProcessorMap1">
    <replaceList>
      <value>valueoffirstiteminmap1</value>
      <value>valueoffseconditeminmap1</value>
      <value>valueoffthirditeminmap1</value>
    </replaceList>
    <replaceList>
      <value>valueoffirstiteminmap2</value>
      <value>valueoffseconditeminmap2</value>
      <value>valueoffthirditeminmap2</value>
    </replaceList>
  </fragment>
  <fragment docname=" Doc2" mapname="BatchProcessorMap2" xpath="
/example1/element1/batch">
    <replaceList>
      <value>valueoffirstiteminmap3</value>
      <value>valueoffseconditeminmap3</value>
    </replaceList>

```

```
</fragment>
</batchProcessor>
```

For this example, consider that:

This XML document has to be structured as the example shown below with batch processor and fragment elements.

The fragment element has three attributes: docname, mapname, xpath. The docname is mandatory, whereas xpath and mapname are optional.

The docname is the name of the XML document which content will be added to the output document.

The mapname is the name of the map that will translate the XML document.

The xpath is applied to the document once it has been translated, and will parse out a document fragment. This document fragment will then be added to the output document.

If value elements exist under the replaceList element, these are used for replacing values in the map. Behind the scenes, these values will be placed in ProcessData as batchProcessingVar1, batchProcessingVar2, etc. The map can then (if desired) pick up these values and replace values during translation of a document. To add this rule to a map, go to Standard Rule. Under “Please enter the Xpath to evaluate”, enter batchProcessingVar1/text() etc.

Doc1 is added to the output document twice, but with different replacement values. Doc2 will then be added once, but only a fragment of the document will be added (since xpath is specified).

Because the InputDocument contains two replaceList under the fragment for Doc1, this document is added twice to the output document. The first time, the map replaces 3 values. The second time, the map replaces the 3 values with different values. The following example shows the Doc1 referenced in the InputDocument and added to the output document:

```
<example1>
  <element1>
    <batch>
      <value>111</value>
      <value>222</value>
      <value>333</value>
    </batch>
    <tag1>tag1</tag1>
    <tag2>tag2</tag2>
    <tag3>tag3</tag3>
  </element1>
</example1>
```

Because the InputDocument contains one replaceList under the fragment for Doc2, this document is added once to the output document. The map will replace 2 values. XPath is applied to the document and the remaining document fragment is added to the output document. The following example shows Doc2 referenced in the InputDocument and added to the output document:

```
<example2>
  <element1>
    <batch>
      <value>111</value>
      <value>222</value>
    </batch>
```

```
        <tag1>tag1</tag1>
        <tag2>tag2</tag2>
        <tag3>tag3</tag3>
    </element1>
</example2>
```

The following example shows the result of invoking the Batch Processor service with the documents:

```
<BatchProcessorOutput>
<example1>
  <element1>
    <batch>
      <value>valueoffirstiteminmap1</value>
      <value>valueoffseconditeminmap1</value>
      <value>valueoffthirditeminmap1</value>
    </batch>
    <tag1>tag1</tag1>
    <tag2>tag2</tag2>
    <tag3>tag3</tag3>
  </element1>
</example1>
<example1>
  <element1>
    <batch>
      <value>valueoffirstiteminmap2</value>
      <value>valueoffseconditeminmap2</value>
      <value>valueoffthirditeminmap2</value>
    </batch>
    <tag1>tag1</tag1>
    <tag2>tag2</tag2>
    <tag3>tag3</tag3>
  </element1>
</example1>
  <batch>
    <value>valueoffirstiteminmap3</value>
    <value>valueoffseconditeminmap3value>
  </batch>
</BatchProcessorOutput>
```

BEA Tuxedo Adapters

The BEA[®] Tuxedo[®] Subscribe adapter maintains communication with a BEA Tuxedo system and subscribes to notifications for all business events between Application and a BEA Tuxedo system.

The BEA Tuxedo Application Services adapter integrates Tuxedo-based applications and real-time transactions, using XML or non-XML EDI formats such as ANSI X12, and supports bidirectional functions for event-driven input and real-time transactions.

The following table provides an overview of the BEA Tuxedo Subscribe adapter and Tuxedo Application Services adapter:

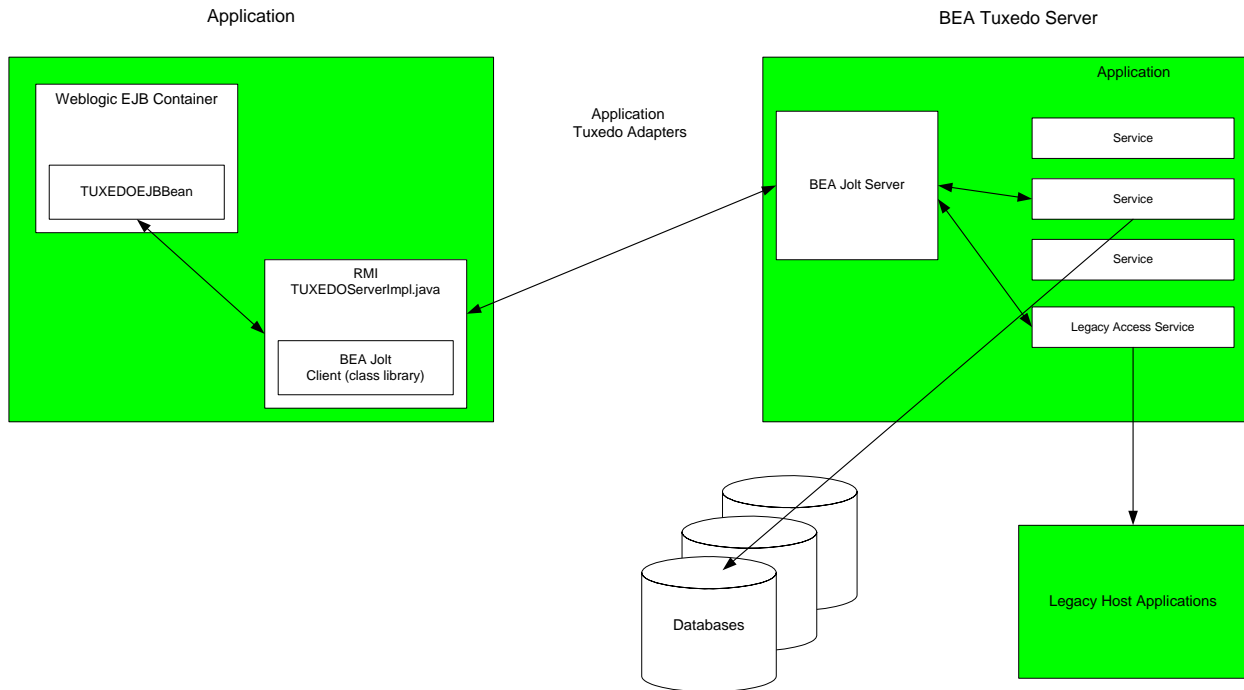
System name	Tuxedo
Graphical Process Modeler (GPM) category	None until you create an application adapter, which will then show up in the Custom category.
Description	<p>The Tuxedo Subscribe adapter and Tuxedo Application Services adapter support BEA Tuxedo release 8.0 or later and offer:</p> <ul style="list-style-type: none">◆ Simple integration of Tuxedo-based applications and real-time transactions, using XML or non-XML EDI formats such as ANSI X12◆ Open communication with Tuxedo-based application services, including asynchronous services (tpacall), synchronous services (tpcall), and message queue (tpqueue) interactions◆ Bidirectional functions for event-driven input and real-time transactions
Business usage	Using BEA Jolt [®] , Enterprise JavaBeans (EJB), and Remote Method Invocation (RMI), Application can connect to a BEA Tuxedo system and request and retrieve Tuxedo services.
Usage example	Used to interact with a banking application to perform typical transactions such as deposit, withdrawal, and inquiry on accounts.
Preconfigured?	No
Requires third party files?	Yes. Before you can install the Tuxedo adapters, the library package BEA Jolt (jolt.jar file) must be available on the computer where the BEA Tuxedo system is installed and on the UNIX host computer where Application is installed. This file placement enables Application to verify corresponding applications and retrieve services from a BEA Tuxedo system.
Platform availability	All supported platforms
Related services	None

Application requirements	Before you can install the Tuxedo adapters, the library package BEA Jolt (jolt.jar file) must be available on the computer where the BEA Tuxedo system is installed and on the UNIX host computer where Application is installed. This file placement enables Application to verify corresponding applications and retrieve services from a BEA Tuxedo system.
Initiates business processes?	Bootstrapping can only be performed by the subscribe adapter for notifications received from Tuxedo. Any application adapters you create do not perform bootstrapping as they are typically transaction driven.
Invocation	No special requirements exist for invocation; simply call the adapter configuration within a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ Success – Tuxedo adapter ran successfully. ◆ Error – Tuxedo adapter encountered an error.
Restrictions	The Tuxedo Subscribe adapter and Tuxedo Application Services adapter support BEA Tuxedo release 8.0 or later only.
Persistence level	System default
Testing considerations	The Tuxedo Server installation comes with a sample application called BankApp that can be used to test this adapter.

How the BEA Tuxedo Adapters Work

Using BEA Jolt®, Enterprise JavaBeans (EJB), and Remote Method Invocation (RMI), Application can connect to a BEA Tuxedo system and request and retrieve Tuxedo services.

The following figure shows how Application interacts with a BEA Tuxedo system:



The following steps summarize how the Tuxedo adapters work within a business process in Application:

1. The Tuxedo Application Services adapter establishes connection between Application and the BEA Tuxedo system, authenticates passwords, and then logs in to the BEA Tuxedo system.
2. Application verifies that the application and its services specified during configuration correspond with the application and services in the BEA Tuxedo system.
3. After verification, the Tuxedo Application Services adapter retrieves and sends the data that makes up the application to Application.
4. Application maps the data retrieved from the BEA Tuxedo system and converts it to Extensible Markup Language (XML).
5. Application builds a Java-based interface based on the XML data, including parameters that represent services, and merges the interface into the GPM.
6. The parameters list options that enable request-and-response transactions for the application.
7. The Tuxedo Subscribe adapter communicates with the BEA Tuxedo system to process notifications as specified.

Tuxedo Subscribe Adapter

Using the BEA Jolt client class library (jolt.jar file), Application can support subscriptions to Tuxedo event-based notifications. When subscribing to event notifications, BEA Jolt internally enables the notification event handler for each event session—thus generating and sending notifications for both

unsolicited and brokered events. Using the Tuxedo Subscribe adapter, you can override the event handler process for the event session. That is, you can disregard the `onReply()` method in the event handler process and define what should happen to all notifications run for the client for that session.

Application supports the following types of event-based notifications:

Unsolicited Event – Generates and sends notifications for all business events, including business events that are broadcast or posted.

Brokered Event – Generates and sends notifications for only specified published business events.

Tuxedo Application Services Adapter

The Tuxedo Application Services adapter enables Application to connect to a BEA Tuxedo system and retrieve previously defined Tuxedo services into Application. Application can then perform the Tuxedo service functions and complete request-and-response transactions defined in business processes.

Implementing the BEA Tuxedo Adapters

To implement the BEA Tuxedo adapters, complete the following tasks:

1. Activate your license for the BEA Tuxedo adapters. See *An Overview of Implementing Services*.
2. Install the adapters. See *Installing the BEA Tuxedo Adapters*.
Note: Both the Tuxedo Subscribe and Tuxedo Application Services adapters are installed together.
3. If subscribing to event-based notifications, create a Tuxedo Subscribe adapter configuration to generate notifications. See *Creating a Service Configuration* and *Configuring the BEA Tuxedo Adapters*.
4. Create a configuration of the Tuxedo Application Services adapter. Configure the adapter to retrieve an application and its services. See *Creating a Service Configuration* and *Application BEA Tuxedo Application Services Adapter Configuration*.
5. Use the Tuxedo Application Services adapter in a business process.
See *Creating a Business Process Using the BEA Tuxedo Application Services Adapter* to define fields that complete request-and-response transactions for Tuxedo adapters.

Installing the BEA Tuxedo Adapters

Before you can install the Tuxedo adapters, the library package BEA Jolt (`jolt.jar` file) must be available on the computer where the BEA Tuxedo system is installed and on the UNIX host computer where Application is installed. This file placement enables Application to verify corresponding applications and retrieve services from a BEA Tuxedo system.

Note: Both the Tuxedo Subscribe and Tuxedo Application Services adapters are installed together.

To install the Tuxedo adapters:

1. Verify that BEA Jolt is installed on the computer where the BEA Tuxedo system is installed.

Note: If you performed a full installation when installing BEA Tuxedo, BEA Jolt is installed. If you did not perform a full installation, you can obtain and install BEA Jolt from either the BEA Tuxedo CD-ROM or from the BEA Web site at <http://commerce.bea.com/downloads/tuxedo.jsp>.

2. In the space below, record the path and name of the folder where BEA Jolt (that is, the folder containing the jolt.jar file) is installed or where you are installing BEA Jolt.

Path: _____

3. From the folder (recorded in step 2) on the BEA Tuxedo system, copy the jolt.jar file to a folder on the UNIX host computer where Application is installed. In the space below, record the path and name of the folder.

Path: _____

4. Shut down Application if it is running.
5. From the folder where Application is installed, set the CLASSPATH environment variable to point to the library package (that is, the jolt.jar file). Use the BEA Tuxedo system version number and the folder path recorded in step 3 to enter a command similar to the following example:

```
./Install3rdParty.sh bea 8_0 -j directory/jolt.jar
```

6. Restart Application.

Configuring the BEA Tuxedo Adapters

To configure the Tuxedo adapters, you must specify field settings in Application and in the GPM. That is:

Configure the Tuxedo Subscribe adapter to subscribe to Tuxedo event-based notifications.

Configure the Tuxedo Application Services adapter for each application and its services for which you want to retrieve from a BEA Tuxedo system.

Application BEA Tuxedo Subscribe Adapter Configuration

Note: The names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Host or IP Address (hostname)	Host name of the BEA Tuxedo system. You can use your IP address in the format 00.000.000.000. Required.
Listening Port (port)	Port name of the BEA Tuxedo system in the format 0000. Required.
User ID (userid)	Name that identifies the authorized user of the BEA Tuxedo application account. Optional.
Password (password)	Password of the authorized user of the BEA Tuxedo application account. Optional.
Application Password (appPsw)	Password of the application for which Application should communicate event-driven notifications. Optional.
Subscription Type	Type of notification to which you are subscribing: <ul style="list-style-type: none"> ◆ Select Subscribe to a named service to receive notifications from a predefined named service in the BEA Tuxedo system when an event occurs. ◆ Select Subscribe to unsolicited notifications to receive notifications when any event occurs.
Published Service Name	If you are subscribing to a predefined service, enter the name of the service in the Published Service Name field. Note: This field is only displayed if you selected Subscribe to a named service as Subscription Type.
Business Process	Specify the business process that Application should run when events occur.
Document Storage Type	This defines how the document will be stored in the system. Required. Valid values are: <ul style="list-style-type: none"> ◆ System Default ◆ Database (default) ◆ File System
User	Type, or select from the list, the user ID to be associated with business process instances of this service.

Application BEA Tuxedo Application Services Adapter Configuration

Because each application from which you want Application to retrieve services requires a separate service configuration, Application provides an exclusive wizard that provides additional fields to configure in addition to the fields used to create a service configuration.

To access and configure the additional fields for the Tuxedo Application Services adapter configuration:

1. Ensure that you have created a service configuration for the Tuxedo Application Services adapter. See *Creating a Service Configuration*.
2. From the **Deployment** menu, select **Adapter Utilities** > BEA Tuxedo.
3. Under BEA Tuxedo **systems**, click **Go!** next to **Launch the BEA Tuxedo Helper Application**.
4. Complete the following fields:

Note: The names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Host or IP Address (hostname)	Host name of the BEA Tuxedo system. You can use your IP address in the format 00.000.000.000. Required.
Listening Port (port)	Port name of the BEA Tuxedo system in the format 0000. Required.
User ID (userid)	Name that identifies the authorized user of the BEA Tuxedo system application account. Optional.
Password (password)	Password of the authorized user of the BEA Tuxedo system application account. Optional.
Application Password (appPsw)	Password of the application from which Application should retrieve services. Optional.
User Role (ACCOUNT_ID)	Role of the user as defined in a BEA Tuxedo system for the application account. Optional.

Field	Description
Application Name	Name of the application to retrieve from the BEA Tuxedo system. Required. For example, to retrieve the banking application distributed with BEA Tuxedo, type BankApp in the Application Name field.
Add Service	Service that represents an action that accomplishes request-and-response transactions within the application. Required. For example, to complete deposit transactions for the BankApp application, type Deposit in the Add Service field. After typing a service name in the Add Service field, click add . Repeat this process until you have added all the services that you want Application to retrieve for this application.
Selected Services (Action)	Displays all the services that you have entered to retrieve from the BEA Tuxedo system. Note: To remove a service from the list, select the service from the Selected Services field, and then click delete .

You are now ready to perform either of the following tasks:

Create another configuration of the Tuxedo Application Services adapter to retrieve another application and services from the BEA Tuxedo system.

Create a business process using the configuration of the Tuxedo Application Services adapter that you just created.

Creating a Business Process Using the BEA Tuxedo Application Services Adapter

To create a business process using a configuration of the Tuxedo Application Services adapter, use the GPM. The GPM displays an icon that uses the name that you indicated when creating the adapter configuration.

When using the adapter configuration to create a business process that interacts with a BEA Tuxedo system, consider that:

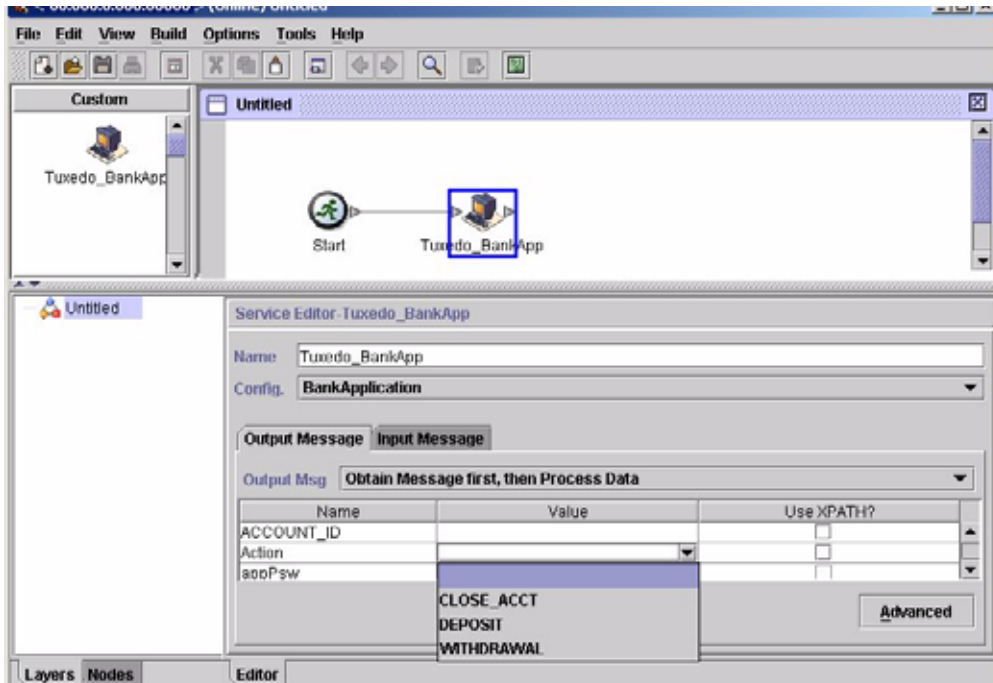
An icon for each adapter configuration that you create using the Tuxedo Application Services adapter displays in the Custom Services stencil.

To view adapter properties, you must drag the icon that represents an adapter configuration to the workspace. Double-click the icon, and select the configuration from the Config list in the Service Editor.

The Action list displays the services request-and-response transactions for business process use.

The following figure shows the GPM. The BEA Tuxedo Application Service adapter has been configured and renamed to Tuxedo BankApp, and an adapter configuration called BankApplication is selected. The

Action parameter displays a list of the services specified during configuration and retrieved from a BEA Tuxedo system.



Begin Transaction Service

The following table provides an overview of the Begin Transaction service.

System name	Begin Transaction Service
Graphical Process Modeler (GPM) categories	All Services
Description	Starts a transaction.
Business usage	Use this service to mark the start of a multi-step transaction in a business process. All of the steps after this service are part of one transaction and are committed or rolled back at the same time. This service should be used with the End Transaction service. The Begin Transaction service can also be used for global transactions where the transaction spans across adapters such as the MQ and Lightweight JDBC adapters.
Usage example	<p>Consider a customer ordering items online and using a shopping cart. The merchant's business process:</p> <ol style="list-style-type: none">1. Gets a document containing shopping cart information (customer information, product details).2. Inserts the customer information to the customer table.3. Inserts product details to the invoice table (product_id, quantity purchased).4. Updates the quantity of the product from the inventory table as the products are sold. <p>If step 4 fails (update the quantity), but steps 1 and step 2 were committed. The merchant's inventory table is now inaccurate, because the items were reported as sold, but the items were not deducted from the inventory table. The Begin Transaction service and End Transaction service make it possible to have a transaction where all the steps are committed successfully or all the steps are rolled back.</p>
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	The Begin Transaction service should be used with the End Transaction service.
Application requirements	Nothing external is required to use this service.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	No

Returned status values	Possible status values: <ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	N/A
Testing considerations	N/A

How the Begin Transaction Service Works

Use the Begin Transaction service to start a multi-step transaction.

Business Process Example

In the following business process example, an online shopping cart is featured. The business process:

1. Gets a document containing shopping cart information (customer information, product details).
2. Inserts the customer information to the customer table.
3. Inserts product details to the invoice table (product_id, quantity purchased)
4. Updates the quantity of the product from the inventory table as the products are sold.

For example, if step 4 fails (update the quantity), but steps 1 and step 2 were committed. The merchant's inventory table is now inaccurate, because the items were reported as sold, but the items were not deducted from the inventory table. The Begin Transaction service and End transaction service make it possible to have a transaction where all the steps are committed successfully or all the steps are rolled back.

```
<process name="TestTran">
  <sequence>
    <operation name="Begin">
      <participant name="BeginTransactionService"/>
      <output message="Xout">
        <assign to="START_TRANSACTION">TRUE</assign>
        <assign to="DISTRIBUTED">TRUE</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="root_element">root</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="LightweightJDBCAdapterType">
```

```

<participant name="LightweightJDBCAdapterQuery"/>
<output message="LightweightJDBCAdapterTypeInputMessage">
  <assign to="schedHour">-1</assign>
  <assign to="result_name">result</assign>
  <assign to="schedDay">-2</assign>
  <assign to="sql">INSERT INTO CUSTOMER_LW (CUSTOMER_ID, SIR_NAME, FIRST_NAME,
MIDDLE_INITIAL, LAST_NAME, ADDRESS, CITY, STATE, ZIP, PHONE_NUM, EMAIL,
ORDERS_PLACED, EMPLOYER, WORK_PHONE, WORK_EMAIL, WORK_ADDRESS1, WORK_ADDRESS2,
WORK_CITY, WORK_STATE, WORK_ZIP) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?) </assign>
  <assign to="param1" from="Testplan/Test4/Customer_ID/text()"></assign>
  <assign to="paramtype1">Integer</assign>
  <assign to="param2" from="Testplan/Test4/SirName/text()"></assign>
  <assign to="paramtype2">String</assign>
  <assign to="param3" from="Testplan/Test4/FirstName/text()"></assign>
  <assign to="paramtype3">String</assign>
  <assign to="param4" from="Testplan/Test4/MiddleInitial/text()"></assign>
  <assign to="paramtype4">String</assign>
  <assign to="param5" from="Testplan/Test4/LastName/text()"></assign>
  <assign to="paramtype5">String</assign>
  <assign to="param6" from="Testplan/Test4/Address/text()"></assign>
  <assign to="paramtype6">String</assign>
  <assign to="param7" from="Testplan/Test4/City/text()"></assign>
  <assign to="paramtype7">String</assign>
  <assign to="param8" from="Testplan/Test4/State/text()"></assign>
  <assign to="paramtype8">String</assign>
  <assign to="param9" from="Testplan/Test4/Zip/text()"></assign>
  <assign to="paramtype9">String</assign>
  <assign to="param10" from="Testplan/Test4/Phone_Num/text()"></assign>
  <assign to="paramtype10">String</assign>
  <assign to="param11" from="Testplan/Test4/Email/text()"></assign>
  <assign to="paramtype11">String</assign>
  <assign to="param12" from="Testplan/Test4/Orders_Placed/text()"></assign>
  <assign to="paramtype12">Integer</assign>
  <assign to="param13" from="Testplan/Test4/Employer/text()"></assign>
  <assign to="paramtype13">String</assign>
  <assign to="param14" from="Testplan/Test4/Work_Phone/text()"></assign>
  <assign to="paramtype14">String</assign>
  <assign to="param15" from="Testplan/Test4/Work_Email/text()"></assign>
  <assign to="paramtype15">String</assign>
  <assign to="param16" from="Testplan/Test4/Work_Address1/text()"></assign>
  <assign to="paramtype16">String</assign>
  <assign to="param17" from="Testplan/Test4/Work_Address2/text()"></assign>
  <assign to="paramtype17">String</assign>
  <assign to="param18" from="Testplan/Test4/Work_City/text()"></assign>
  <assign to="paramtype18">String</assign>
  <assign to="param19" from="Testplan/Test4/Work_State/text()"></assign>
  <assign to="paramtype19">String</assign>
  <assign to="param20" from="Testplan/Test4/Work_Zipcode/text()"></assign>
  <assign to="paramtype20">String</assign>
  <assign to="pool">mysqlPool</assign>
  <assign to="row_name">row</assign>
  <assign to="schedMinute">-1</assign>
  <assign to="query_type">ACTION</assign>
  <assign to="." from="*"></assign>
</output>

```



```

    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="LightweightJDBCAdapterType">
    <participant name="LightweightJDBCAdapterQuery"/>
    <output message="LightweightJDBCAdapterTypeInputMessage">
      <assign to="schedHour">-1</assign>
      <assign to="result_name">result</assign>
      <assign to="schedDay">-2</assign>
      <assign to="sql">INSERT INTO INVOICE CUSTOMER_ID,PRODUCT_ID,QTY) VALUES (?, ?,
?) </assign>
      <assign to="param1" from="Testplan/Test4/Customer_ID/text()"></assign>
      <assign to="paramtype1">Integer</assign>
      <assign to="param2" from="Testplan/Test4/PRODUCT_ID/text()"></assign>
      <assign to="paramtype2">String</assign>
      <assign to="param3" from="Testplan/Test4/QUANTITY/text()"></assign>
      <assign to="paramtype3">Integer</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="LightweightJDBCAdapterType">
    <participant name="LightweightJDBCAdapterQuery"/>
    <output message="LightweightJDBCAdapterTypeInputMessage">
      <assign to="schedHour">-1</assign>
      <assign to="result_name">result</assign>
      <assign to="schedDay">-2</assign>
      <assign to="sql">INSERT INTO PRODUCT QTY VALUES (?) WHERE PRODUCT_ID=?
</assign>
      <assign to="param1" from="Testplan/Test4/QUANTITY/text()"></assign>
      <assign to="paramtype2">Integer</assign>
      <assign to="param3" from="Testplan/Test4/PRODUCT_ID/Text()"></assign>
      <assign to="paramtype3">String</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation>
    <participant name="EndTransactionService"/>
    <output message="Xout">
      <assign to="END_TRANSACTION">TRUE</assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

Implementing the Begin Transaction Service

To implement the Begin Transaction service for use in a business process:

1. Use the Begin Transaction service in a business process.
2. Use the End Transaction service to end the transaction.

Note: When a pool is defined in `jdbc.properties`, you must mark the pool being referenced as transactional:

```
myPool.transactional=true
```

Output from Business Process to Service

The following table contains the parameter passed from the business process to the Begin Transaction service:

Parameter	Description
Start Transaction	Starts the transaction. Must be set to TRUE for processing to be transactional. Valid value is true (default) or false. Required.
Distributed	Controls whether or not the transaction is distributed or extended to adapters with transactional support. For additional information, see <i>Listing of Services and Adapters by type</i> for a list of adapters with transactional support. Valid value is true or false (default). Optional.
On Fault	Controls the action to take when an error occurs. To undo the transaction to the point of error, select ROLLBACK. To keep the work done to the point of error to occur, select COMMIT. Valid value is Rollback or Commit (default). Optional.

BP Exception Service

The following table provides an overview of the BP Exception service:

System name	BPEXceptionService
Graphical Process Modeler (GPM) categories	All Services, System
Description	Enables you to specify an error and have a business process raise a named error indicating that a fault has occurred in a business process.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	No
Application requirements	None
Initiates business processes?	No
Invocation	Not applicable
Business process context considerations	None
Returned status values	None
Restrictions	No
Testing considerations	None

How the BP Exception Service Works

The BP Exception service sets the basic status of a business process to **ERROR** and sets the suggested `errorCode` parameter as the advanced status. The `onFault` mechanism uses the advanced status to determine if a handler is triggered. The service enables you to set a status report, as well.

Business Process Example

The following figure shows one example of how the BP Exception service could be used in a business process:

```
<operation name="generateException">
  <participant name="BPEXceptionService"/>
  <output message="Xout">
    <assign to="errorCode">foo:timeout</assign>
    <assign to="." from="*"></assign>
  </output message>
</operation>
```

```
</output>
<input message="Xin">
  <assign to="." from="*"></assign>
</input>
</operation>
```

Implementing the BP Exception Service

To implement the BP Exception service, complete the following tasks:

1. Create a BP Exception service configuration. For information, see *Creating a Service Configuration*.
2. Configure the BP Exception service.
3. Use the B2B Exception service in a business process.

Configuring the BP Exception Service

To configure the BP Exception service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
exceptionCode	The advanced status to raise. If no exceptionCode is supplied, the default is the code si:general.
statusReport	String to use as the status report for the service. Optional.

BP Fault Log Adapter

The following table provides an overview of the BP Fault Log adapter:

System name	BPFaultLog
Graphical Process Modeler (GPM) categories	All Services, System
Description	Provides control over business process error message logging and recovery.
Business usage	The following options are available with the BP Fault Log adapter: <ul style="list-style-type: none">◆ Logging business process error messages◆ Setting a logging level – DEBUG, INFO, WARN, and ERROR◆ Including the document body in the logged message◆ Causing a failure in the business process
Usage example	The BP Fault Log adapter logs business process error messages and assigns log levels to various types of messages. Specifying a log level helps categorize and prioritize the generated logs. You may find it useful to configure the BP Fault Log adapter to fail a business process if a service introduces invalid data, and to log a message indicating the source of the bad data. This feature can save you time and help you identify problems within your business process so you can correct them. Specify a location on disk to send and view generated business process error message logs.
Preconfigured?	No
Requires third party files?	Text editor (such as Microsoft WordPad) to view the logs.
Platform availability	All supported platforms
Related services	None
Application requirements	Designate a location on disk to send and view logs.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	When set to fail on error, it will set the WFC status to ERROR.
Returned status values	Success, Error
Restrictions	None
Persistence level	None
Testing considerations	The BP Fault Log adapter uses system defaults to terminate logger lines.

How the BP Fault Log Adapter Works

The BP Fault Log adapter logs business process error messages and assigns log levels to various types of messages. Specifying a log level helps categorize and prioritize the generated logs. You may find it useful to configure the BP Fault Log adapter to fail a business process if a service introduces invalid data, and to log a message indicating the source of the bad data. This feature can save you time and help you identify problems within your business process so you can correct them. Specify a location on disk to send and view generated business process error message logs.

Note: The BP Fault Log adapter terminates logger lines with 0A (linefeed). Use a text editor (for example, Microsoft WordPad) to view the log file.

Implementing the BP Fault Log Adapter

To implement the BP Fault Log adapter, complete the following tasks:

1. Create a BP Fault Log adapter configuration. For information, see *Creating a Service Configuration* on page 15.
2. Configure the BP Fault Log adapter.
3. Use the B2B Fault Log adapter in a business process.

Configuring the BP Fault Log Adapter

To configure the BP Fault Log adapter, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the BP Fault Log adapter in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a group	Select one of the three options available for service groups: <ul style="list-style-type: none">◆ None – this service will not be part of a service group.◆ Create New Group – to create a new service group for this service type, enter a unique name.◆ Select Group – select an existing service group for this service type from the list.
Logger Name (loggerName)	Name of the log. Required. The file created in the Logger Output Path is given this name and an extension of .log.

Field	Description
Logger Output Path (loggerPath)	Path used to write the log to a file. Required. Valid value is a valid full path.
Include Document Body (includeBody)	Whether to include the body of the data with the error message. Required. Valid values are: <ul style="list-style-type: none"> ◆ True – Include the error message and the body of the data. ◆ False – Include the error message only.
Logging Type	Select type of log. Required. Valid values are Normal and Rolling.

GPM Configuration

The following table describes the fields used to configure the BP Fault Log adapter in the GPM:

Field	Description
Config	Name of the adapter configuration.
fail-on-error	Whether to cause the business process to fail if an error is logged. Optional. Valid values are Y and N. If the parameter is set to Yes, and the log-level (see next field description) is set to ERROR, the business process status is set to Error and the business process advanced status is set to the value in the log-message. This enables you to force a business process to fail (based on invalid business process data), when it may have ordinarily succeeded.
log-level	Type of log. Required. Values are: <ul style="list-style-type: none"> ◆ DEBUG – An error message for internal use ◆ INFO – Who the error message was sent to ◆ WARN – A significant error in the business process ◆ ERROR – A general error in the business process <p>Note: You can add custom log levels to this list in the GPM, or by editing a business process in the business process editor.</p>
log-message	Message that you want to send to the log. Required.
maxLogSize	Maximum log size for rolling in bytes. Required if you selected Rolling as the log type in the service configuration. Valid value is a number greater than zero.

BP Fault XML Log adapter

The following table provides an overview of the BP Fault XML Log adapter:

System name	BPFaultLogXML
Graphical Process Modeler (GPM) category	All Services
Description	Enables a business process writer to log business specific messages to a file in XML format. Each log item is an XML document fragment.
Business usage	Use this service to provide a custom log to a business process. This can also be used to read a created log for formatting or processing.
Usage example	You have written a large BPML to process widgets, and would like to provide feedback to a log to track the processing or any faults that occur. The XML generated can be read for formatting. A typical read example would be a read and format into HTML using XSLT.
Preconfigured?	No
Requires third party files?	None
Platform availability	All supported platforms
Related services	None
Application requirements	Designate a location on disk to send and view logs.
Initiates business processes?	No
Invocation	Must be a step in the business process or in the on-fault handler.
Business process context considerations	For WRITE operations: When set to fail on error, the business process context status will be set to ERROR.
Returned status values	Success, Error Note: For WRITE operations, this adapter will always return a status of "Success" because it is typically a non-critical part of a business process. If a failure occurs it is logged, but the business process continues.
Restrictions	None
Persistence level	Default
Testing considerations	None

Implementing the BP Fault XML Log adapter

To implement the BP Fault XML Log adapter, complete the following tasks:

1. Create a BP Fault XML Log adapter configuration.

2. Configure the BP Fault XML Log adapter.
3. Use the BP Fault XML Log adapter in a business process.

Configuring the BP Fault XML Log adapter

To configure the BP Fault XML Log adapter, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration.
action	<p>Valid values are:</p> <ul style="list-style-type: none"> ◆ READ—Reads the specified log from the specified directory, and uses the loggerName value as a prefix of the log file to read and loggerPath value as the directory path of where the log files are located. ◆ WRITE—performs a normal log write operation.
combineLog	<p>Valid values are:</p> <ul style="list-style-type: none"> ◆ True – Logs will be combined. When more than one log file name matches the prefix specified by the loggerName value, these log files will be combined into well-formed XML and placed onto process data as a single document. The process data document name is the specified loggerName value. ◆ False – Logs are not combined. When more than one log file name matches the prefix specified by the loggerName value, each log file will be placed onto process data. The process data document names are the same as the original log file names.
fail-on-error	<p>Whether to cause the business process to fail if an error is logged. Optional. Valid values are True and False. Used for WRITE operations only.</p> <p>If this parameter is set to True, and the log-level parameter is set to ERROR, the business process status is set to Error and the business process advanced status is set to the value in the log-message. This enables you to force a business process to fail (based on invalid business process data), when it may have ordinarily succeeded.</p>
includeBody	<p>Whether to include the body of the data with the error message. Required. Used for WRITE operations only.</p> <p>When the body is included, only the first 2048 bytes are included.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ True – Include the error message and the body of the data. ◆ False – Include the error message only.
log-level	<p>Type of log. Required. Used for WRITE operations only. Values are:</p> <ul style="list-style-type: none"> ◆ DEBUG – An error message for internal use ◆ ERROR – A general error in the business process ◆ INFO – Who the error message was sent to ◆ WARN – A significant error in the business process <p>Note: You can add custom log levels to this list in the GPM, or by editing a business process in the business process editor.</p>

Field	Description
log-message	Message that you want to send to the log. Required. Used for WRITE operations only.
loggerName	Name of the log. Required. The file created in the Logger Output Path is given this name and an extension of .log. Used for both READ and WRITE operations.
loggerPath	Path used to write the log to a file. Required. Used for both READ and WRITE operations. Valid value is a valid full path.
logType	Select type of log. Required. Used for WRITE operations only. Valid values are Normal and Rolling.
maxLogSize	Maximum log size for rolling in bytes. Required if you selected Rolling as the log type in the service configuration. Used for WRITE operations only. Valid value is a number greater than zero.

Business Process Example

The following example shows how the BP Fault XML Log adapter could be used in a business process:

```
<process name="default">
  <operation name="BPFaultLogXML">
    <participant name="mybpfaultxmllog"/>
    <output message="BPFaultLogXMLInputMessage">
      <assign to="action">WRITE</assign>
      <assign to="combineLog">>false</assign>
      <assign to="fail-on-error">>true</assign>
      <assign to="includeBody">>true</assign>
      <assign to="log-level">DEBUG</assign>
      <assign to="log-message">hello log</assign>
      <assign to="loggerName">testlog.log</assign>
      <assign to="loggerPath">/home/user/</assign>
      <assign to="logType">normal</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

BP Metadata Info Service

The following table provides an overview of the BP Metadata Info service:

System name	Business Process Metadata
Graphical Process Modeler (GPM) category	All Services
Description	<p>Use this service to retrieve information about a particular business process such as:</p> <ul style="list-style-type: none">◆ Business process definition ID◆ Process name◆ Description◆ State◆ Status◆ Type◆ Priority◆ Persistence level◆ Lifespan◆ Business process definition version◆ Storage type◆ Recovery level◆ Doc tracking flag◆ Deadline◆ Event level◆ Sub/parent business process information◆ Correlations <p>Note: This service can also be used to retrieve information about the business process that runs this service.</p>
Business usage	Enables you to retrieve information about a business process. This information can then be made available to the system to be used in other processes.
Usage Example	See <i>Business Process Example</i> .
Preconfigured?	Yes. The preconfigured copy is called BPMetadataInfoService.
Requires third party files?	No
Platform availability	All supported platforms
Related services	None

Application requirements	None
Initiates business processes?	No
Invocation	Does not invoke a business process.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	None
Persistence level	None
Testing considerations	None

Implementing the BP Metadata Info Service

To implement the BP Metadata Info service, complete the following tasks:

1. Create a BP Metadata Info service configuration.
2. Configure the BP Metadata Info service.
3. Use the BP Metadata Info service in a business process.

Configuring the BP Metadata Info Service

To configure the BP Metadata Info service you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
LINKAGE	Enable links between this BP and any parent/child processes. Optional. Valid values are True and False. Default is True.
DISPOSITION	Print out the WFD_STATE and WFD_STATUS information to Process Data. Optional. Valid values are True and False. Default is True.
WORKFLOW_ID	Specify a workflow ID from which to get MetaData. Optional; Reports on itself if no other process is specified.
TRACE	Dump all the WorkFlow Context information. Optional. Valid values are True and False. Default is False.
CORRELATION	Report on correlations. Optional. Valid values are True and False. Default is False.

Output from Service to Business Process

The following table contains the parameters passed from the BP Metadata Info service to the business process:

Parameter	Description
WORKFLOW_ID	The current Workflow ID. Required. Specify a known instance, or obtain from process data such as if using this service in an on-fault. Default is current workflow ID.
MESSAGE_FROM_SERVICE	The user who send the message from service. Optional. Default is current user.
WFD_ID	The current WorkFlow Definition ID. Required.
WFD_VERSION	Current BP version. Required.
WFD_NAME	Current BP name. Required.
WFD_DESCRIPTION	Current BP description. Required.
WFD_STATE	Current BP state; only show if the disposition is set. Optional.
WFD_STATUS	Current BP status; only show if the disposition is set. Optional.
WFD_TYPE	Current BP type. Required.
WFD_PRIORITY	Current BP priority. Required.
WFD_PERSISTENCE_LEVEL	Current BP persistence level. Required.
WFD_LIFE_SPAN	Current BP lifespan. Required.
WFD_STORAGE_TYPE	Current Storage type. Required.
WFD_RECOVERY_LEVEL	Current Recovery level. Required.
WFD_DOC_TRACKING_FLAG	Current Document tracking flag. Required.
WFD_DEADLINE_INTERVAL	Current Deadline interval. Optional.
WFD_EVENT_LEVEL	Current Event Level. Optional.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the BP Metadata Info service:

Parameter	Description
WORKFLOW_ID	Specify a workflow ID from which to get MetaData. Optional; Reports on itself if no other process is specified.

Business Process Example

The following example business process illustrates using the BP Metadata Info service:

```
<process name="BPMetadataInfoService_01">
  <sequence name="simple">
    <operation name="1">
      <participant name="BPMetadataInfoService" />
      <output message="Xout">
        <assign to="WORKFLOW_ID">9</assign>
        <assign to="DISPOSITION">>true</assign>
        <assign to="LINKAGE">>true</assign>
        <assign to="CORRELATION">>true</assign>
        <assign to="TRACE">>true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

BP Report Service

The following table provides an overview of the BP Report service:

System name	BP Report Service
Graphical Process Modeler (GPM) categories	All Services
Description	<p>Use the BP Report service to generate reports that contain a list of business process instance IDs that are not longer running and their states match the state you specify in the business process configuration. You can specify the following states of the business processes in the business process configuration:</p> <ul style="list-style-type: none">◆ Halting◆ Active◆ Waiting_on_io
Business usage	<p>Use this service to generate a list of business processes in one or more of the following states:</p> <ul style="list-style-type: none">◆ Halting◆ Active◆ Waiting_on_io
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	N/A
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ <code>wfc.setBasicStatus(WorkFlowContext.SUCCESS)</code>◆ <code>wfc.setBasicStatus(WorkFlowContext.ERROR)</code>
Restrictions	N/A
Persistence level	System Default

Output from Business Process to Service

The following table contains the parameters passed from the business process to the BP Report service:

Field	Description
STATE	Specify the state of the business process for which you want to generate a report. Required. Valid values are: <ul style="list-style-type: none">◆ Halting◆ Active◆ Waiting_on_io The default value is Active.
showReport	View the details in the status report, such as, the queue the business process was or is in.
termInvalidWFD	Terminate a business process in one of the following states when the business process definition no longer exists in the system: <ul style="list-style-type: none">◆ halted◆ interrupted_auto◆ interrupted_man

Business Process Example

The following example business process illustrates using the BP Report service:

```
<operation name="2">
  <participant name="BPReportService"/>
  <output message="Xout">
    <assign to="STATE">active,halting</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

Cache Refresh Service

The following table provides an overview of the Cache Refresh service:

System name	CacheRefreshServiceType
Graphical Process Modeler (GPM) categories	All Services, Process Controls
Description	The Cache Refresh service allows a user to refresh cached customer defined property file information without restarting the Application system.
Business usage	As a performance enhancement, you may write a business process that will pick up and cache information from a customer-defined properties file. When the same data is frequently referenced, caching it saves the time that would have been used for repeated lookups. The Cache Refresh service allows updated property file information in the cache to be refreshed without a restart of the system.
Usage example	<p>For business reasons, information, such as a fax number, needs to be included in output documents. This information can be stored in the properties file and cached. When this information needs to be changed, such as a new fax number in this example, after the properties file has been changed, the cache must be refreshed or the system restarted. This service can be used to refresh the cache without a system restart.</p> <p>Note: Refreshing the cache makes updated property file information instantly available to all instances of business processes that reference it.</p>
Preconfigured?	Yes. A CacheRefreshService configuration is available. However, values for required fields need to be entered when the Cache Refresh service is used in a business process.
Requires third party files?	No
Platform availability	All supported platforms
Related services	None
Application requirements	None
Initiates business processes?	No.
Invocation	The Cache Refresh service is invoked by a business process.
Business process Considerations	The properties file needs to be referenced in the business process using an XPath function (sci-get-property). To run this service on a schedule, create a business process containing this service and then schedule the business process.
Returned status values	None
Restrictions	The referenced customer property file must either exist in the <Application>/properties directory, or the <Application>/properties/servers.properties file must contain a tag/value pair with "FileTag=<the full file description>". Only customer property files can be refreshed using this service. Cached Application property file information should not be updated while Application is running.

Persistence level	System Default
Testing considerations	None

Cache Refresh Service Operation

When invoked, the Cache Refresh service:

1. Flushes the cache of all tag/value pair information from the property file referenced by the *PropertyFileTag* parameter in the sci-get-property XPath function
2. Reads the property file
3. Stores all tag/value pair information in the cache

The Cache Refresh service searches for the *PropertyFileTag* property file in the following manner:

1. The service looks in the properties directory of the Application installation for the *PropertyFileTag.properties* property file. If found, the service reads the property file.
2. The service looks in the properties directory of the Application installation for the *PropertyFileTag* (*exact match*) file.
3. The service reads the servers.properties file in the properties directory of the Application installation. If a tag for the *PropertyFileTag* property is found in the servers.properties file, the Cache Refresh service reads the property file.
4. If the *PropertyFileTag* property file is not found, the service displays an error message.

Note: The sci-get-property XPath function will read a property file into cache if it references a *PropertyFileTag* parameter that is not already in the cache. After the new *PropertyFileTag* and its contents are initially loaded into the cache, the Cache Refresh service must be run before the new information will be available in the cache.

Implementing the Cache Refresh Service

To implement the Cache Refresh service, complete the following tasks:

1. Configure the Cache Refresh service.
2. Use the Cache Refresh service in a business process.

Configuring the Cache Refresh Service

To configure the Cache Refresh service, you must specify settings for the following fields in the Graphical Process Modeler (GPM):

Field	Description
Config	Unique and meaningful name for the service configuration. Required.

Field	Description
cache_name	Name of the property file or the property file tag. Required. Valid value: any string. Note: This name must match the PropertyFileTag in sci-get-property. For more information, see <i>Referencing the Properties File</i> .
cache_type	Type of cache to use. Required. Valid value: Properties

Output from Business Process to Service

The following table contains the parameters passed from the business process to the Cache Refresh service:

Field	Description
cache_name	Name of the property file or the property file tag. Required. Valid value: any string.
cache_type	Type of cache to use. Required. Valid value: Properties

Business Process Examples

The following example business process illustrates the use of the Cache Refresh service:

```
<operation>
  <participant name="CacheRefreshService"/>
  <output message="Xout">
    <assign to="cache_type" from="'properties'"/></assign>
    <assign to="cache_name" from="'customer_foo'"/></assign>
  </output>
  <input message="Xin">
  </input>
</operation>
```

Referencing the Properties File

XPath Function Syntax:

```
sci-get-property("PropertyFileTag", "PropertyName")
```

Where:

PropertyFileTag – The name of the tag defined in servers.properties or the property file name in the properties directory

PropertyName – The name of the property in the properties file

Example:

A property file named *customer_fax.properties* has been created that contains the following assign statements:

```
description = some_fax_information
some_fax_number=000.555.1212
```

Place the *customer_fax.properties* file in the <Gentran Integration Suite installation>/properties directory. This file is referenced by the PropertyFileTag parameter and the tag/value pair contents of this file are read into the cache.

After the property tag *customer_fax* has been created, the following XPath function statements can be used to retrieve the data from the *customer_fax.properties* file:

```
<assign to="SomeDescription" from="sci-get-property('customer_fax','description')"/>  
<assign to="faxNumber" from="sci-get-property('customer_fax','some_fax_number')"/>
```

CDATA Conversion Service

The CDATA Conversion service is designed to be used only with the Transora™ Data Catalog (TDC) 3.2 adapter. The following table provides an overview of the CDATA Conversion service:

System name	CDATAConversionService
Graphical Process Modeler (GPM) categories	All Services, Transora
Description	Converts CDATA sections within the primary document to and from PCDATA. This service is used only with the Transora (TDC) 3.2 adapter.
Business usage	Transora does not accept XML documents with CDATA sections. Certain Application services, such as the Translation service, may require any XML documents to use entity characters (& > < ' ") in CDATA sections rather than using XML entity references such as & for the ampersand character. This service converts the CDATA used by Application to the Transora-required PCDATA format as needed.
Usage example	A user wants to send an Add Item request to Transora, and they use a Translation service with a map to generate the Transora XML request document. The XML output of the Translation service contains CDATA sections, but Transora will not accept CDATA. So the CDATAConversionService is used to convert the CDATA sections to PCDATA with entity references.
Preconfigured?	Yes. A configuration named CDataConversionService is installed with the TDC 3.2 adapter.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	TDC 3.2 adapter
Application requirements	No
Initiates business processes?	No
Invocation	Not applicable
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – Primary document was successfully converted.◆ Error – CDataMode parameter was not set, or the primary document could not be parsed and converted.
Restrictions	None. Any number of instances may be created, but typically a single instance can be shared by any number of business processes.

Testing considerations

To test it, create an instance, then run it from a business process as described in the usage example. Provide an XML document as the primary document for the business process. Viewing the resulting document in a browser from the Application web console can be problematic; XML entity references and CDATA sections may not appear in the browser the same way they are stored in the underlying document. Therefore, a better method is to save the resulting file to disk and view it using a text editor or XML editing tool.

How the CDATA Conversion Service Works

The CDATA Conversion service converts CDATA sections within the primary document to and from PCDATA.

Outbound

In CDATA_TO_PCDATA mode, Application converts any CDATA sections to PCDATA, and converts the five standard entity characters (&, <, >, “, and ‘) to their corresponding entity references (& < > " '). For example, see the following fragment:

```
<myTag><![CDATA[Joe & Sally]]></myTag>
```

would be converted to:

```
<myTag>Joe &amp; Sally</myTag>
```

Inbound

In PCDATA_TO_CDATA mode, the primary document must contain a valid XML document. In PCDATA_TO_CDATA mode, elements containing any of the five standard entity references are converted to CDATA sections without entity references. For example, & converts to the single character &. The following example illustrates this conversion:

```
<myTag>Joe &amp; Sally</myTag>
```

would be converted to:

```
<myTag><![CDATA[Joe & Sally]]></myTag>
```

Code Sample

The following code example runs the CDATA Conversion service in a business process:

```
<operation name="PCDATA to CDATA">
  <participant name="CDataConversionService"/>
  <output message="noopout">
    <assign to="." from="*"></assign>
    <assign to="CDataMode">PCDATA_TO_CDATA</assign>
  </output>
  <input message="noopin">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

Implementing the CDATA Conversion Service

The CDATA Conversion service installs automatically with the TDC 3.2 adapter, and is part of the TDC 3.2 adapter implementation. For information, see *Transora Data Catalog (TDC) 3.2 Adapter*.

For general information about creating service and adapter configurations, see *Managing Services and Adapters*.

To implement the CDATA Conversion service, complete the following tasks:

1. Implement and configure the TDC 3.2 adapter to be used with this service. For information, see *Transora Data Catalog (TDC) 3.2 Adapter*.
2. Configure the pre-installed copy of the CDATA Conversion service. For information, see *Configuring the CDATA Conversion Service* on page 95.
3. Use the CDATA Conversion service in a business process.

Configuring the CDATA Conversion Service

To configure the CDATA Conversion service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.
CDataMode	Data conversion mode for the adapter. Optional at configuration time, but required at run time. Valid values are PCDATA_TO_CDATA and CDATA_TO_PCDATA.

Certificate Revocation List (CRL) Processing Service

The following table provides an overview of the CRL Processing service:

System name	CRLProcessingService
Graphical Process Modeler (GPM) category	All Services
Description	<p>This service processes a version 2 digital Certificate Revocation List (CRL). Specifically, this service:</p> <ul style="list-style-type: none">◆ Marks those certificates in the Application database which appear on the list as revoked or held◆ Checks the cache and notifies you that a specific certificate is revoked when an attempt is made to check in that system or trusted certificate. <p>Note: The notification only occurs if you enable CRL caching in Application.</p>
Business usage	<p>Use this service to confirm that your business processes correctly authenticate your trading partners.</p> <p>This service confirms that other Application services that request digital certificates are only given valid certificates. Valid certificates are those that have not been revoked or held.</p> <p>Also use this service if you want to be notified when an attempt is made to check in a revoked certificate.</p>
Usage example	<p>The CRL Processing service should be placed in a scheduled business process that downloads CRLs at the required interval. (The required interval varies by authority and trading community.)</p> <p>Note: The CRL does not contain certificates; it lists those certificates that should be revoked.</p> <p>For more information, see <i>How the CRL Processing Service Works</i></p>
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The HTTP Client adapter and the LDAP adapter are often used in conjunction with this service because these adapters can be used to download the CRL.</p> <p>This service confirms that other services which request the use of digital certificates are given them only if the certificate has not been revoked or is being held.</p>
Application requirements	The list passed to this service must be formatted according to the CRL version 2 specification published by the Internet Engineering Task Force (IETF) in document http://www.ietf.org/rfc/rfc3280.txt .
Initiates business processes?	No

Invocation	This service must be preceded by a service or adapter that can collect the CRL and hand it off to this one. In addition, the entire chain of authorization for the CRL must be in the Application CA certificate store before this service can run.
Business process context considerations	<p>This service processes the CRL; it does not download the CRL.</p> <p>Before processing the CRL, this service attempts to verify the signature on the CRL. This service can verify the CRL only if the following certificates are in the Application certificate store:</p> <ul style="list-style-type: none"> ◆ The certificate used to sign the CRL ◆ Any intermediate certificates between that certificate and the CA root certificate ◆ The CA root issuing certificate
Returned status values	<ul style="list-style-type: none"> ◆ Success – The CRL processed correctly. ◆ Error – The CRL did not process correctly. <p>For a complete description of the errors, see the log file.</p>
Restrictions	<p>This service only processes the following CRL critical extensions:</p> <ul style="list-style-type: none"> ◆ crlNumber ◆ baseCRLNumber ◆ IssuingDistributionPoints.
Persistence level	None
Testing considerations	<p>Enable the ShowTranscripts parameter in the CRL Processing service configuration. When enabled, operations on each certificate in the list are recorded.</p>

How the CRL Processing Service Works

The Certificate Revocation List is periodically issued by the certificate authority to identify those certificates that should be revoked.

The Certificate Revocation List Processing service takes a CRL as input and does the following:

1. It verifies the CRL using a certificate in the CA certificate store in the Application database.
2. Once verified, it checks the CRL expiration date. If the CRL is expired, the default behavior is for the service to fail. You can allow the service to continue by changing the FailOnExpiredCRL parameter to false.
3. The service then begins to process the CRL. If the CRL has a critical extension that the application does not understand, the default behavior for this service is to fail. You can allow the service to continue by changing the FailOnUnsupportedCRLCritExtns parameter to false.
4. For each entry in the CRL, the service looks in the trusted certificate and system certificate tables for a corresponding certificate entry.

Note: You can turn off processing of the system certificates by adding the parameter RevokePrivateKeys to the BMPL and setting it to false.

For each record in the system certificate or trusted certificate table which corresponds to an entry in the CRL, the service does the following:

- ◆ If the status of the entry is OK and the CRL indicates the certificate is revoked, the service sets the status in the table to revoked.
- ◆ If the status of the entry is OK and the CRL indicates the certificate is held, the service sets the status in the table to held.

When the status is set to revoked or held, the API calls which attempt to load the certificate or key from the database will fail at runtime.

- ◆ If the status of the entry is held and the CRL indicates the certificate should be removed from the CRL, the service sets the status in the table to OK. This allows the API calls which attempt to load the certificate or key from the database to succeed at runtime.

5. The service then determines if it is a base CRL or a delta CRL and adds it to the Application database according to the following rules:

- ◆ If there is no corresponding CRL of that type and issuer, the service simply adds the CRL to the database.
- ◆ If there is an existing CRL, the service uses the CRL number to add the most recent CRL to the database.
 - If the CRL numbers are identical, the service uses the thisUpdate date in the CRLs to determine which is the most recent.
 - If there are no CRL numbers, the thisUpdate date is used.

6. The service updates the CRL cache.

The CRL cache is a standard cache that maintains CRL information in memory and is checked by Application when you try to upload a certificate and by the certificate loading APIs. The cache consists of information from one base CRL and possibly one delta CRL for each issuer.

Note: CRLs are verified when they are loaded into the cache. By default, CRLs that are expired or that have unsupported critical extensions do not update the CRL cache. However, you can change the default by setting the LoadExpiredCRLs and LoadUnsuppCritExtnCRLs parameter to true in the security.properties file.

Implementing the CRL Processing Service

To implement the CRL Processing service, complete the following tasks:

1. Activate your license for the CRL Processing service. See *Installing Application*.
2. Create a CRL Processing service configuration. See *Managing Services and Adapters*.
3. Configure the CRL Processing service.
4. Use the CRL Processing service in a business process.

Configuring the CRL Processing Service

To configure the CRL Processing service, you must specify settings for the following fields in the GPM.

Note: You must use the Advanced button in the GPM Service Editor to set the values for the fields listed in the following table. These fields do not automatically appear in the Service Editor. The default values are used if these fields are not defined in the GPM.

Field	Description
Config	Name of the service configuration. Required.
ShowTranscripts	Tells the service whether to put log information into the WFC advanced status. For debug information, debug logging must be enabled in the security log. Optional. Valid values are true and false (default is false).
FailOnExpiredCRL	Tells the service whether to fail if the CRL is expired. The only reason to not fail in this situation is for testing. Optional. Valid values are true and false (default is true).
FailOnUnsupportedCRLCritExtns	Tells the service whether to fail if it encounters a critical extension it does not understand when processing a CRL. In most cases, Application should not process CRLs with un-supported critical extensions. There may be extraordinary or testing situations where you want to turn this off. Optional. Valid values are true and false (default is true).

Output from Business Process to Service

The preceding table describes the information that is passed from the business process to the CRL Processing service.

Business Process Example

The following example illustrates how to pass values to the service for the parameters in the previous table:

```
<process name="lCRL">
  <sequence name="optional">
    <operation name="One">
      <participant name="lCRL"/>
      <output message="noopout">
        <assign to="." from="*"></assign>
        <assign to="ShowTranscripts">true</assign>
        <assign to="FailOnExpiredCRL">>false</assign>
        <assign to="FailOnUnsupportedCRLCritExtns">>false</assign>
      </output>
      <input message="noopin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Check Expire Service

The Check Expire service alerts you when certificates are about to expire, or have expired.

Note: You cannot check the expiration for the OpsDrv, OpsKey, and UIKey certificates. These system certificates are used internally by Application and do not expire.

The following table provides an overview of the Check Expire service:

System Name	Check Expire Service
Graphical Process Modeler (GPM) category	All Services
Description	Use this service to retrieve information on all supported certificates that have expired or will expire in a set number of days. The number of days can be set by changing the checkexpireDays value in ui.properties.
Business usage	Alerts you when certificates are about to expire, or have expired.
Usage example	Can be used in a business process to report on any certificates that are about to expire in the number of days specified by the EXPIRE_DAYS parameter.
Preconfigured?	Yes. A configuration of the service called CheckExpireService is installed with Application.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	The Check Expire service is included in a predefined business process, Schedule_CheckExpireService.bp, which is scheduled by default to run at 2:30 a.m. daily. To see whether there are any certificates about to expire, view messages under Accounts > User News.
Business process context considerations	None
Returned status values	None
Restrictions	The OpsDrv, OpsKey, and UIKey certificates cannot be checked for expiration because they are used internally by Application and do not expire.
Persistence level	None
Testing considerations	None

How the Check Expire Service Works

The Check Expire service runs on a schedule (by default daily at 2:30 a.m.) and looks at each checked in certificate to compare its expiration date against the value set in Application (the default is 14 days). Any certificates that have already expired or will expire sometime within the specified number of days are listed on the Admin Console Home page in System Alerts, System Alerts on Dashboard, and the Event Viewer under the Operator menu. The alerts remain in the system for 60 days.

Implementing the Check Expire Service

There is no implementation necessary for the Check Expire service, unless you want to change the number of days prior to expirations that you receive alerts, or the schedule. To do so, edit the service configuration called Check Expire Service. The following table describes the fields used to configure the Check Expire adapter in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select a Service Group to associate with this adapter. Valid values: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time. Default.◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.◆ For more information about service groups, see <i>Managing Services and Adapters</i>.
Run as User	Type the user ID to associate with the schedule, or select a user ID from the list. Valid value is any valid Application user ID.
Use 24 Hour Clock Display	If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.

Field	Description
Schedule	<p>By default, the Check Expire service and business process are scheduled to run every day at 2:30 a.m.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.
Expire Days	Number of days before a certificate expires that you will begin receiving alerts. Default is 14. Required.
Output Method	Type of alert to use when a certificate is about to expire. Valid values are Email, EVENT and User Message.
Output Email Address (Output Method = Email)	E-mail address to use when the alert output method is Email.
User(s) Settings (Output Method = User Message)	Users for whom an alert message will be created when a certificate is about to expire. Used when the alert output method is User Message.
Exclusion Settings	Certificates to exclude from reporting.

Output from Service to Business Process

The following table contains the parameters passed from the Check Expire service to a business process:

CHECKEXPIRE	Specifies the type of alert to use when a certificate is about to expire. Valid values are Email, EVENT and User Message.
EXPIRE_DAYS	Specifies when you start receiving alerts that a certificate is about to expire. For example, if this parameter is set to 7 days, you will start receiving alerts seven days before the certificate is scheduled to expire. Default is 14.

Business Process Example

This is the Schedule_CheckExpireService business process as installed with Application:

```
<process name="Schedule_CheckExpireService">
  <sequence>
    <operation name="Service">
      <participant name="CheckExpireService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Command Line Adapter

The Command Line adapter is a second generation adapter that enables Application to run a program from a command line in a business process. This includes executable programs, scripts, or operating system (OS) commands external to Application.

The Command Line adapter operates in a remote implementation only. This does not necessarily mean that it has to run remotely. It runs in a separate JVM (Java Virtual Machine), which may be on the machine where Application is installed or on a remote machine.

The following table provides a high-level overview of the Command Line adapter:

System name	CmdLine
GPM category	All Services
Description	Executes a program from the command line. The syntax is: <code>cmd.exe /C <command></code> This is not necessary when running scripts. Examples: <code>cmd.exe /C dir</code> <code>importBPs.sh</code>
Business usage	Used to call any program from the command line. Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the Command Line Adapter is in the process of being replaced by the Command Line Adapter 2.
Usage example	You could use the Command Line adapter to invoke a program that: <ul style="list-style-type: none">◆ Encrypts and decrypts data you want to send or receive securely over the Internet◆ Manipulates data, such as change every occurrence of one letter to another◆ Pages someone◆ Initiates a business process◆ Initiates a remote system These are just a few examples out of many possible uses.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Command Line Adapter 2
Application requirements	None
Initiates business processes?	Yes, if you define a business process to start when you configure the Command Line adapter. The business process starts after the output from the command line process is read.

Invocation	No special requirements. The Command Line adapter can either be used to start (“bootstrap”) a business process or you can include the Command Line adapter directly in a business process to perform an explicit command. Note: The term “bootstrap” is used in the GPM to indicate that the Command Line adapter is used to start a business process after the output from the command line process is read.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ Success: Command Line adapter was successful. ◆ Error: Command Line adapter was unsuccessful.
Restrictions	A configuration of this adapter is needed for each program invoked from the command line.
Persistence level	System default (Full Persistence)
Testing considerations	Call a small command line process (without using it to invoke a business process) to perform a simple command.

How the Command Line Adapter Works

Use the Command Line adapter in a business process to run any program from the command line, including executable programs, scripts, or OS commands external to Application. The types of activities that can be performed include data encryption and decryption, file manipulation, data manipulation, and initiation of a process on a remote system.

You can create multiple Command Line adapter configurations, one for each of several specific commands. Alternatively, you can use a single Command Line adapter configuration to perform different commands by specifying the command line process (`cmdLine`) and working directory (`workingDir`) in the business process. See *Command Line* for details on these parameters.

For example, your company communicates with a legacy database that is important to its daily business. You want to retrieve some customer billing information in the database and send it within a business process in Application to your accounting department. You can write your own executable program to communicate with your legacy system and run it using the Command Line adapter.

The following steps summarize how the Command Line adapter is typically used in a business process:

1. The adapter writes the content of the current primary document to a file in the working directory specified as the value of the working directory parameter. The name of this file is specified by the value of the `inputFile` parameter.
2. Application runs an executable program that picks up the file and sends it to the legacy system.
3. The legacy system returns a file, which now includes the customer billing information, and the adapter retrieves it. The file returned is specified by the value of the `outputName` parameter.
4. The adapter reads the file contents into the primary document.
5. Application performs the next operation in the business process.

Implementing the Command Line Adapter

You can implement a Command Line adapter to do the following:

Execute commands using the command line from within a business process.

Invoke the Command Line adapter on a schedule and then start a new business process using the output from the adapter.

Note: This could be used if you wanted to schedule a command line program that accessed a legacy database on a regular schedule and then used the output in a business process.

The information in this section applies to both of the above implementations.

Before You Begin

Before you begin to implement the Command Line adapter, complete the following tasks:

1. Create and test the command line program or command to make sure that it works.
2. Determine the working directory where you will be processing your commands.

Process Overview

To implement the Command Line adapter, complete the following tasks:

1. Create a Command Line adapter configuration. See *Managing Services and Adapters*.
2. Configure the Command Line adapter. See *Configuring the Command Line Adapter*.
3. Create and enable a business process that includes the Command Line adapter.

Note: If you are configuring a Command Line adapter to start a business process, create the business process before configuring the adapter.

4. Test the business process and the adapter.
5. Run the business process.

Configuring the Command Line Adapter

To create a Command Line adapter configuration, you must specify field settings in Application and in the GPM. For general information about service and adapter configurations, see *Managing Services and Adapters*.

Application Configuration


The following table describes the fields used to configure the Command Line adapter in Application.

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference. Some fields can be configured in the GPM, if not selected here. Regardless of where they are configured, they can be overridden using BPML.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.

Field	Description
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select a Service Group to associate with this adapter.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. Default. ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: A Service Group is a group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. For more information about Service Groups, see <i>Managing Services and Adapters</i>.</p>
Remote Name (remoteName)	<p>Remote host name or IP address where the remote adapter implementation is running. Required. Default value is localhost.</p> <p>If you have an instance of the Command Line adapter that uses the rmiAddr parameter, this parameter must be updated with the name of the machine where CLA2Client.jar is running.</p>
Remote Port (remotePort)	<p>Remote port that the remote adapter implementation is listening on. Required. Default value: Port on which the CLA2Client.jar was autostarted, normally basePort+52.</p> <p>If you have an instance of the Command Line adapter that uses the rmiAddr parameter, this parameter must be updated with the port number where CLA2Client.jar is running.</p>

Field	Description
Command Line (cmdLine)	<p>Command line process you want to run. Do one of the following:</p> <ul style="list-style-type: none"> ◆ If you want to set this parameter in the GPM/business process, leave the field blank. ◆ Type the command line process in this field exactly as you would from the command line. ◆ If you want to use a command that redirects input or output (through the use of >, <, or), you must do so using a script file. ◆ If you do not know the input or output file name, type the following parameters in the command line process as placeholders: <ul style="list-style-type: none"> • \$Input • \$Output <p>These parameters are typed directly in the command line process. You can use these parameters on the command line in any order and multiple times if necessary. At run time, they are replaced with the actual file name.</p> ◆ If you want to enter user parameters, use the following placeholders: \$0 – \$9. These placeholders are resolved by the parm0 – parm9 parameters defined in the GPM or using BPML. <p>Note: If \$Input or \$Output resolves to a filename that contains one or more spaces, automatic quoting will be performed before the command line is executed. For example, if the original command line was <code>test.sh \$Input</code>, and \$Input resolves to file 1, then the final command line, before execution, will be <code>test.sh "file 1"</code>. Therefore, do not put quotes around \$Input or \$Output.</p> <p>Note: An example of a command line entry is <code>test.sh \$Input \$Output \$0 \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9</code>. This runs the shell script <code>test.sh</code> taking an input file, using ten parameters, and producing an output file.</p>
Working Directory (workingDir)	<p>Location of the directory to use for executing the command line process. Optional. Default is the current working directory of the JVM running CLA2Client.jar.</p> <p>Caution: Using this adapter to call a Unix script modifies the directory path of the environment variable LD_LIBRARY_PATH. To keep your current path, your script should include either the LD_LIBRARY_PATH path or a reference to your .profile (which includes the LD_LIBRARY_PATH path).</p> <p>Note: The CLAClient.jar is no longer used.</p>
Turn on debugging messages? (cla2_debug)	<p>Turn on debugging for this adapter instance?</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Logging is turned on and the messages are written to the system log. ◆ No (false) – Default. <p>Note: This turns on debugging for this specific adapter instance. These messages are logged in the system log in the <code>install_dir/logs</code> directory. This parameter is read-only in the GPM.</p>

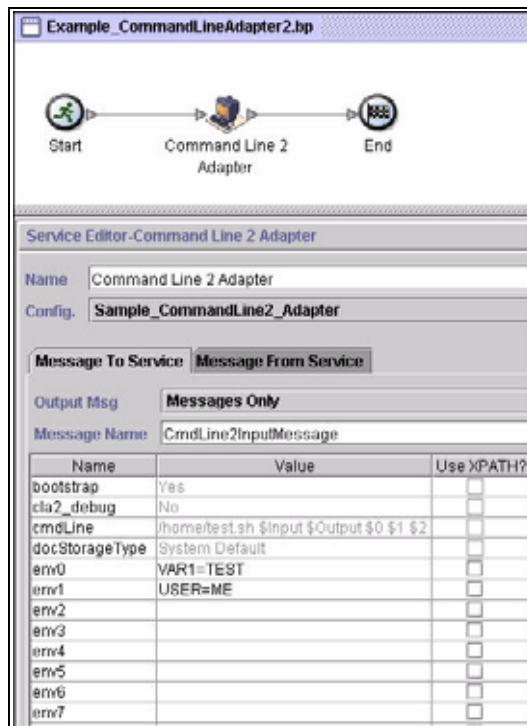
Field	Description
Wait on the process to complete before continuing? (waitOnProcess)	<p>Wait on the process to complete before continuing the business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – If the value is Yes, a status report is created if any stdout/stderr is generated by the process. If an error occurs while the service is processing output data, the advanced status contains the error message instead of the return code value. ◆ No (false) <p>Note: If <i>Use the output generated by the command line process</i> is set to Yes, the value of this parameter is assumed to be Yes because the service cannot use output if it does not wait for the process to complete. This parameter is read-only in the GPM.</p>
Does this service start a business process? (bootstrap)	<p>Whether the service starts a business process. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) ◆ No (false) <p>Note: This parameter is read-only in the GPM.</p>
Business process (initialWorkFlowName)	<p>Business process you want the Command Line adapter to start. This field is required only if you selected Yes in <i>Does this service start a business process?</i>. If you prefer to configure this parameter in the GPM, select Not Applicable.</p>
Document Storage Type (docStorageType)	<p>Defines how the document is stored in the system. Required when the adapter starts a business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ System Default – Default ◆ Database ◆ File System <p>Note: For more information on document storage types, see <i>Managing Services and Adapters</i>.</p>
Run as User	<p>Applies to the scheduling of the business process. The Run As User field only displays as an option if <i>Does this service start a business process?</i> is set to Yes.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid value is any valid Application user ID.</p> <p>Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>
Use 24 Hour Clock Display	<p>If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.</p>

Field	Description
Schedule	<p>Information about scheduling the business process invoked by the Command Line adapter. The Schedule field only displays as an option if <i>Does this service start a business process?</i> is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.
Does the command line process require an input file? (useInput)	<p>Defines whether the command line process requires an input file? Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – The primary document of the current business process context is written out to the file system in the working directory and is used as input to the process. Default. ◆ No (false) – No file is written to disk even if a document exists in the business process context. <p>Note: This parameter is read-only in the GPM.</p>
Input File Name (inputName)	<p>Input file name, if the command line process requires an input file. Any occurrences of \$Input in the command line are replaced with this name.</p> <p>Optional. If you leave this field blank, the default is the primary document name.</p> <p>Note: It is important to have a unique input file name for all concurrently running instances of Command Line adapters. If more than one instance of the Command Line adapter can be executing at the same time, you must create a dynamic, unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine.</p>

Field	Description
Delete input file after process completes? (inputDelete)	<p>Defines whether the input file is deleted after the process completes?</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Default ◆ No (false) <p>Note: To delete the input file, <i>Wait on the process to complete before continuing?</i> must also be Yes. This parameter is read-only in the GPM.</p>
Use the output generated by the command line process? (useOutput)	<p>Use output generated by the command line process? Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – The adapter will attempt to read the output of the process. If bootstrapping a workflow, the file will become the primary document in the new workflow. If not bootstrapping, the file will become the primary document of the current workflow. Default. ◆ No (false) – No file is read into the business process context even if one is generated by the command line process. <p>Note: This parameter is read-only in the GPM.</p>
Output File Name (outputName)	<p>Output file name, if you want to use the output generated by the command line process. Any occurrences of \$Output in the command line are replaced with this name. Optional. If you leave this field blank, the default is the business process primary document name.</p> <p>Note: It is important to have a unique output file name for all concurrently running instances of command line adapters. If more than one instance of the Command Line adapter can be executing at the same time, you must create a dynamic unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine.</p>
Delete output file after process completes? (outputDelete)	<p>Specifies whether the output file is deleted after it is collected?</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Default ◆ No (false) <p>Note: This parameter is read-only in the GPM.</p>

GPM Configuration

The following screen shows a graphical view of some GPM parameters for the Command Line adapter. The dimmed values were specified using the Command Line adapter configuration. The active fields are env0 and env1, which cannot be configured in the service configuration.



The following example shows the corresponding business process solution using BPML.

```
<process name="Example_CommandLineBP">
  <operation name="Command LineAdapter Run Script">
    <participant name="Sample_CommandLine_Adapter"/>
    <output message="CmdLineInputMessage">
      <assign to="." from="*" />
      <assign to="parm0">VAR1</assign>
      <assign to="parm1">USER</assign>
      <assign to="parm2">10</assign>
      <assign to="env0">VAR1=TEST</assign>
      <assign to="env1">USER=ME</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```


The following table describes the fields used to configure the Command Line adapter in the GPM. This table contains the fields that are only configured in the GPM. Other fields may also be configured if they were left blank in the Application configuration.

Field	Description
Config (participant name)	Name of the adapter configuration. Required.
env0	An environment variable in the form name=value. Optional. Any value is valid.
env1	An environment variable in the form name=value. Optional. Any value is valid.
env2	An environment variable in the form name=value. Optional. Any value is valid.
env3	An environment variable in the form name=value. Optional. Any value is valid.
env4	An environment variable in the form name=value. Optional. Any value is valid.
env5	An environment variable in the form name=value. Optional. Any value is valid.
env6	An environment variable in the form name=value. Optional. Any value is valid.
env7	An environment variable in the form name=value. Optional. Any value is valid.
env8	An environment variable in the form name=value. Optional. Any value is valid.
env9	An environment variable in the form name=value. Optional. Any value is valid.
keepPath	Normally, any path information is stripped off the filename to allow for platform independence. This parameter allows you to keep the entire path. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes – Path information is retained ◆ No – Path information is stripped off
parm0	Resolves the \$0 placeholder. Optional. Any value is valid.
parm1	Resolves the \$1 placeholder. Optional. Any value is valid.
parm2	Resolves the \$2 placeholder. Optional. Any value is valid.
parm3	Resolves the \$3 placeholder. Optional. Any value is valid.
parm4	Resolves the \$4 placeholder. Optional. Any value is valid.
parm5	Resolves the \$5 placeholder. Optional. Any value is valid.
parm6	Resolves the \$6 placeholder. Optional. Any value is valid.
parm7	Resolves the \$7 placeholder. Optional. Any value is valid.
parm8	Resolves the \$8 placeholder. Optional. Any value is valid.
parm9	Resolves the \$9 placeholder. Optional. Any value is valid.

Field	Description
setSoTimeout	<p>Specifies, in milliseconds, how long the socket will wait in receive mode without receiving anything before timing out. This is necessary to ensure that a process doesn't "hang" indefinitely. Optional.</p> <p>Valid value: any integer. Default is 60000 milliseconds (60 seconds).</p> <p>If your command line process is going to take longer than the default 60 seconds to process completely, then increase this value accordingly.</p>
successValue	<p>If waitOnProcess is Yes (true), then this option can be used to determine what the successful return code value is. Optional.</p> <p>Valid value is any integer. The default is 0.</p> <p>If a value is specified and does not equal the return code value of the process, the business process status is set to ERROR.</p> <p>Note: The successValue parameter is an important parameter that is often overlooked. It is used to signal Application if the command line process failed. If the returned success value does not match the returned status, the process fails. Without returning a success value from an OS script, failures are not detected and the process is assumed to have passed. This creates a failure for the business functionality that is hard to correct later. In writing OS scripts, always check the return status for each call and handle it properly. This includes returning the status values to the OS shell. Error handling in scripts can cause the script to exit before the final output file is generated. Returning from the script to Application without an output file is a critical error that is handled before the returned successValue is examined. See <i>Use the output generated by the command line process?</i> for dealing with this issue.</p> <p>Many OS commands do not return a success value, instead they output errors to stderr or stdout. In these cases, the commands stderr and/or stdout text must be captured, filtered, and an error status returned if the command failed.</p>

Output from Adapter to Business Process

The following table contains the parameters passed from the Command Line adapter to the business process:

Parameter Name and Element Value (BPML)	Description
Document (CLA/document)	If a file is collected in non-bootstrap mode, the document is placed in ProcessData, not as the Primary Document.
DocumentId (CLA/documentId)	If a file is collected in non-bootstrap mode, the document identifier of the document is placed here.
ProcessExitValue (CLA/ProcessExitValue)	Sets the process data value to the exit value of the process.
FileName (CLA/FileName)	The name of the file, if any, that was collected as part of the output from the process that ran.

Usage Examples

This section contains an example using the Command Line adapter. Examples are included using both the GPM and BPML.

Invoking the Command Line Adapter to Run a Shell Script

The following example business process illustrates using the Command Line adapter to execute a shell script that expects an input file as the first parameter, an output file as the second parameter, and three other parameters.

When this example configuration is used, a shell script called “test.sh” (which resides in the /home directory) is run.

The program requires the input filename as the first parameter, the output filename as the second parameter, and three other parameters.

Because the useInput variable is set to true and the inputName variable is blank, the name of the primary document replaces the \$Input placeholder.

Because the useOutput variable is set to true and the outputName variable is blank, the \$Output placeholder is replaced with the name of the primary document.

If the document name in the workflow context is “data.txt” in this example, the command line becomes `/home/test.sh data.txt data.txt VAR1 USER 10` at run-time.

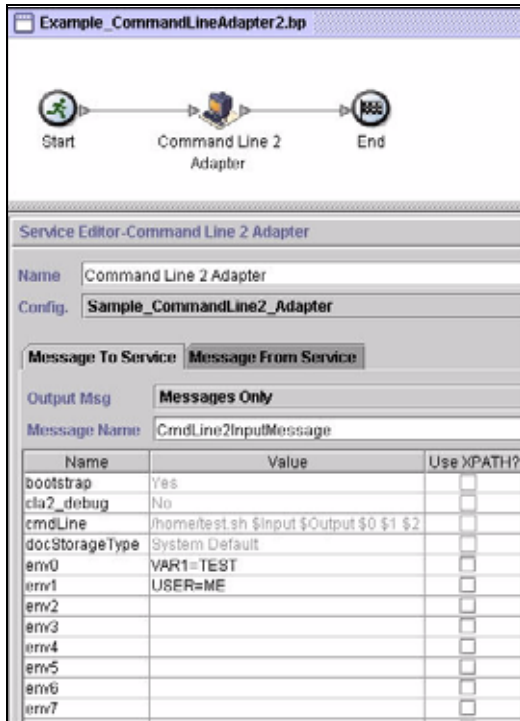
The name of the primary document is passed as the input file to the shell script program on the command line.

The name of the primary document is passed as the output file to the shell script program on the command line.

Note: If the inputName and outputName parameters had file names entered, these file names would replace the \$Input and \$Output placeholders.

GPM Example

The following example illustrates the above business process using the GPM.



Business Process Modeling Language (BPML) Example

The following example illustrates the same business process using BPML.

```
<process name="Example_CommandLine_BP">
  <operation name="Command Line Adapter Run Script">
    <participant name="Sample_CommandLine_Adapter"/>
    <output message="CmdLineInputMessage">
      <assign to="." from="*" />
      <assign to="parm0">VAR1</assign>
      <assign to="parm1">USER</assign>
      <assign to="parm2">10</assign>
      <assign to="env0">VAR1=TEST</assign>
      <assign to="env1">USER=ME</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

Invoking the Command Line Adapter

Caution: If you are utilizing remote invocations of the Command Line adapter or have instances that are using the `rmiAddr` instance parameter, you must manually replace all deployed instances of the `CLAClient.jar` file with the `CLA2Client.jar`.

Adapter startup at the time of Application startup is driven by (Windows) `startWindowsService.cmd` or (UNIX and iSeries) the `run.sh` script.

To start the Command Line adapter:

1. Locate the client `.jar` file (`CLA2Client.jar`) that contains all the necessary classes. This is located in the `install_dir/client/cmdlinedirectory`.
2. If the adapter is going to be run on a different machine, copy the `CLA2Client.jar` file to the system that will run the remote adapter.

Note: The `CLA2Client.jar` does not need to be moved if the remote version is on the same machine.

3. Start the remote adapter, enter:

```
java -jar CLA2Client.jar <port
```

Example: `ava -jar CLA2Client.jar 15699`

The *port* can be any port that is not in use. This is the value to assign in the Remote Port field in the service configuration. See *Remote Port (remotePort)*. If you upgrade Application at any time, you will also need to use the corresponding new `CLA2Client.jar` file to avoid a “ClassConflict” error.

Note: You may also run `CLA2Client.jar` with these commands:

- ◆ (UNIX or iSeries) `./startCmdLine2.sh`
- ◆ (Windows service) `startCLA2WindowsService.cmd`

Stopping the Command Line Adapter

If Application is shut down with (Windows) `stopWindowsService.cmd` or (UNIX and iSeries) `hardstop.sh` script, the Command Line adapter also shuts down. You can also stop the Command Line Adapter 2 with these commands:

```
(UNIX or iSeries) ./stopCmdLine2.sh
```

```
(Windows service) stopCLA2WindowsService.cmd
```

Otherwise, once started, the adapter runs silently as configured and will not return to the command line until it is finished, interrupted, or fails. Therefore, you cannot use that command line to execute any other commands.

Command Line Adapter 2

The Command Line Adapter 2 is a second generation adapter that enables Application to run a program from a command line in a business process. This includes executable programs, scripts, or operating system (OS) commands external to Application.

The Command Line Adapter 2 operates in a remote implementation only. This does not necessarily mean that it has to run remotely. It runs in a separate JVM (Java Virtual Machine), which may be on the machine where Application is installed or on a remote machine.

The Command Line Adapter 2 supports large files up to 12 GB and provides better memory allocation than the Command Line adapter. The Command Line Adapter 2 will eventually replace the Command Line adapter.

The following table provides a high-level overview of the Command Line Adapter 2:

System name	CmdLine2
GPM category	All Services
Description	Executes a program from the command line. The syntax is: <code>cmd.exe /C < ></code> This is not necessary when running scripts. Examples: <code>cmd.exe /C dir</code> <code>importBPs.sh</code>
Business usage	Used to call any program from the command line.
Usage example	You could use the Command Line Adapter 2 to invoke a program that: <ul style="list-style-type: none">◆ Encrypts and decrypts data you want to send or receive securely over the Internet◆ Manipulates data, such as change every occurrence of one letter to another◆ Pages someone◆ Initiates a business process◆ Initiates a remote system These are just a few examples out of many possible uses.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Command Line Adapter
Application requirements	None
Initiates business processes?	Yes, if you define a business process to start when you configure the Command Line Adapter 2. The business process starts after the output from the command line process is read.

Invocation	No special requirements. The Command Line Adapter 2 can either be used to start (“bootstrap”) a business process or you can include the Command Line Adapter 2 directly in a business process to perform an explicit command. Note: The term “bootstrap” is used in the GPM to indicate that the Command Line Adapter 2 is used to start a business process after the output from the command line process is read.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ Success: Command Line Adapter 2 was successful. ◆ Error: Command Line Adapter 2 was unsuccessful.
Restrictions	A configuration of this adapter is needed for each program invoked from the command line.
Persistence level	System default (Full Persistence)
Testing considerations	Call a small command line process (without using it to invoke a business process) to perform a simple command.

How the Command Line Adapter 2 Works

Use the Command Line Adapter 2 in a business process to run any program from the command line, including executable programs, scripts, or OS commands external to Application. The types of activities that can be performed include data encryption and decryption, file manipulation, data manipulation, and initiation of a process on a remote system.

You can create multiple Command Line Adapter 2 configurations, one for each of several specific commands. Alternatively, you can use a single Command Line Adapter 2 configuration to perform different commands by specifying the command line process (cmdLine) and working directory (workingDir) in the business process. See *Command Line* on page 122 for details on these parameters.

For example, your company communicates with a legacy database that is important to its daily business. You want to retrieve some customer billing information in the database and send it within a business process in Application to your accounting department. You can write your own executable program to communicate with your legacy system and run it using the Command Line Adapter 2.

The following steps summarize how the Command Line Adapter 2 is typically used in a business process:

1. The adapter writes the content of the current primary document to a file in the working directory specified as the value of the working directory parameter. The name of this file is specified by the value of the inputFile parameter.
2. Application runs an executable program that picks up the file and sends it to the legacy system.
3. The legacy system returns a file, which now includes the customer billing information, and the adapter retrieves it. The file returned is specified by the value of the outputName parameter.
4. The adapter reads the file contents into the primary document.
5. Application performs the next operation in the business process.

Implementing the Command Line Adapter 2

You can implement a Command Line Adapter 2 to do the following:

Execute commands using the command line from within a business process.

Invoke the Command Line Adapter 2 on a schedule and then start a new business process using the output from the adapter.

Note: This could be used if you wanted to schedule a command line program that accessed a legacy database on a regular schedule and then used the output in a business process.

The information in this section applies to both of the above implementations.

Before You Begin

Before you begin to implement the Command Line Adapter 2:

1. Create and test the command line program or command to make sure that it works.
2. Determine the working directory where you will be processing your commands.

Process Overview

To implement the Command Line Adapter 2:

1. Create a Command Line Adapter 2 configuration. For information, see *Managing Services and Adapters*.
2. Configure the Command Line Adapter 2. For information, see *Configuring the Command Line Adapter 2* on page 120.
3. Create and enable a business process that includes the Command Line Adapter 2.

Note: If you are configuring a Command Line Adapter 2 to start a business process, create the business process before configuring the adapter.

4. Test the business process and the adapter.
5. Run the business process.

Configuring the Command Line Adapter 2

To create a Command Line Adapter 2 configuration, you must specify field settings in Application and in the GPM. For general information about service and adapter configurations, see *Managing Services and Adapters*.


The Application Configuration

The following table describes the fields used to configure the Command Line Adapter 2 in Application.

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference. Some fields can be configured in the GPM, if not selected here. Regardless of where they are configured, they can be overridden using BPML.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select a Service Group to associate with this adapter.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. Default. ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: A Service Group is a group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. For more information about Service Groups, see <i>Managing Services and Adapters</i>.</p>
Remote Name (remoteName)	<p>Remote host name or IP address where the remote adapter implementation is running. Required.</p> <p>Note: For backward compatibility, the CLA2 supports the Command Line adapter parameter rmiAddr (at the business process level only).</p>
Remote Port (remotePort)	<p>Remote port that the remote adapter implementation is listening on. Required. Default value: Port on which the CLA2Client.jar was autostarted, typically basePort+52.</p>

Field	Description
Command Line (cmdLine)	<p>Command line process you want to run. Do one of the following:</p> <ul style="list-style-type: none"> ◆ If you want to set this parameter in the GPM/business process, leave the field blank. ◆ Type the command line process in this field exactly as you would from the command line. ◆ If you want to use a command that redirects input or output (through the use of >, <, or), you must do so using a script file. ◆ If you do not know the input or output file name, type the following parameters in the command line process as placeholders: <ul style="list-style-type: none"> • \$Input • \$Output <p>These parameters are typed directly in the command line process. You can use these parameters on the command line in any order and multiple times if necessary. At run time, they are replaced with the actual file name.</p> ◆ If you want to enter user parameters, use the following placeholders: \$0 – \$9. These placeholders are resolved by the parm0 – parm9 parameters defined in the GPM or using BPML. <p>Note: If \$Input or \$Output resolves to a filename that contains one or more spaces, automatic quoting will be performed before the command line is executed. For example, if the original command line was <code>test.sh \$Input</code>, and \$Input resolves to file 1, then the final command line, before execution, will be <code>test.sh "file 1"</code>. Therefore, do not put quotes around \$Input or \$Output.</p> <p>Note: An example of a command line entry is <code>test.sh \$Input \$Output \$0 \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9</code>. This runs the shell script <code>test.sh</code> taking an input file, using ten parameters, and producing an output file.</p>
Working Directory (workingDir)	<p>Location of the directory to use for executing the command line process. Optional. Default is the current working directory of the JVM running CLA2Client.jar.</p> <p>Caution: Using this adapter to call a Unix script modifies the directory path of the environment variable LD_LIBRARY_PATH. To keep your current path, your script should include either the LD_LIBRARY_PATH path or a reference to your .profile (which includes the LD_LIBRARY_PATH path).</p>
Turn on debugging messages? (cla2_debug)	<p>Turn on debugging for this adapter instance?</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Logging is turned on and the messages are written to the system log. ◆ No (false) – Default. <p>Note: This turns on debugging for this specific adapter instance. These messages are logged in the system log in the <code>install_dir</code> logs directory. This parameter is read-only in the GPM.</p> <p>Note: For backward compatibility, the CLA2 supports the Command Line adapter parameter <code>cmdl_debug</code> (at the business process level only).</p>

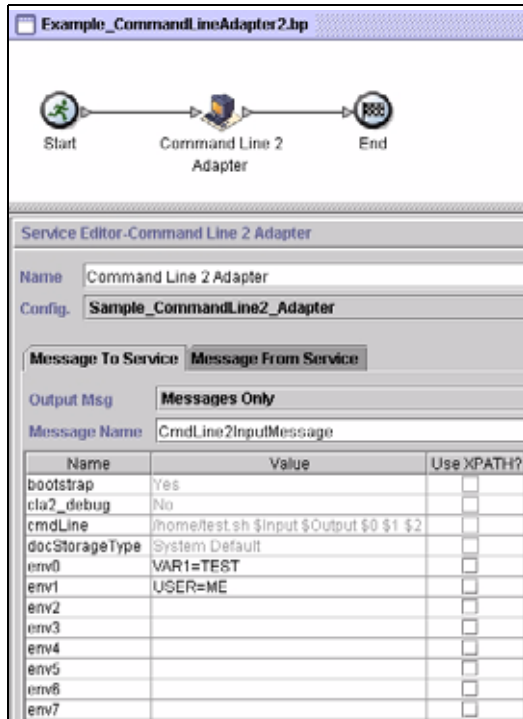
Field	Description
Wait on the process to complete before continuing? (waitOnProcess)	<p>Wait on the process to complete before continuing the business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – If the value is Yes, a status report is created if any stdout/stderr is generated by the process. If an error occurs while the service is processing output data, the advanced status contains the error message instead of the return code value. ◆ No (false) <p>Note: If <i>Use the output generated by the command line process</i> is set to Yes, the value of this parameter is assumed to be Yes because the service cannot use output if it does not wait for the process to complete. This parameter is read-only in the GPM.</p>
Does this service start a business process? (bootstrap)	<p>Whether the service starts a business process. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) ◆ No (false) <p>Note: This parameter is read-only in the GPM.</p>
Business process (initialWorkflowName)	<p>Business process you want the Command Line Adapter 2 to start. This field is required only if you selected Yes in <i>Does this service start a business process?</i> on page 123. If you prefer to configure this parameter in the GPM, select Not Applicable.</p> <p>Note: For backward compatibility, the CLA2 supports the Command Line adapter parameter initialWorkflowId (at the business process level only).</p>
Document Storage Type (docStorageType)	<p>Defines how the document is stored in the system. Required when the adapter starts a business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ System Default – Default ◆ Database ◆ File System <p>Note: For more information on document storage types, see <i>Managing Services and Adapters</i>.</p>
Run as User	<p>Applies to the scheduling of the business process. The Run As User field only displays as an option if <i>Does this service start a business process?</i> is set to Yes.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid value is any valid Application user ID.</p> <p>Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>
Use 24 Hour Clock Display	<p>If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.</p>

Field	Description
Schedule	<p>Information about scheduling the business process invoked by the Command Line Adapter 2. The Schedule field only displays as an option if <i>Does this service start a business process?</i> on page 123 is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.
Does the command line process require an input file? (useInput)	<p>Defines whether the command line process requires an input file? Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – The primary document of the current business process context is written out to the file system in the working directory and is used as input to the process. Default. ◆ No (false) – No file is written to disk even if a document exists in the business process context. <p>Note: This parameter is read-only in the GPM.</p>
Input File Name (inputName)	<p>Input file name, if the command line process requires an input file. Any occurrences of \$Input in the command line are replaced with this name.</p> <p>Optional. If you leave this field blank, the default is the primary document name.</p> <p>Note: It is important to have a unique input file name for all concurrently running instances of Command Line adapters. If more than one instance of the Command Line Adapter 2 can be executing at the same time, you must create a dynamic, unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine.</p>

Field	Description
Delete input file after process completes? (inputDelete)	<p>Defines whether the input file is deleted after the process completes?</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Default ◆ No (false) <p>Note: To delete the input file, <i>Wait on the process to complete before continuing?</i> must also be Yes. This parameter is read-only in the GPM.</p>
Use the output generated by the command line process? (useOutput)	<p>Use output generated by the command line process? Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – The adapter will attempt to read the output of the process. If bootstrapping a workflow, the file will become the primary document in the new workflow. If not bootstrapping, the file is collected and placed in ProcessData, not as the Primary Document. Default. For example, <code><assign name="Assign" to="PrimaryDocument" from="CLA2/document/@SCIOBJECTID"></assign></code> ◆ No (false) – No file is read into the business process context even if one is generated by the command line process. <p>Note: This parameter is read-only in the GPM.</p>
Output File Name (outputName)	<p>Output file name, if you want to use the output generated by the command line process. Any occurrences of \$Output in the command line are replaced with this name. Optional. If you leave this field blank, the default is the business process primary document name.</p> <p>Note: It is important to have a unique output file name for all concurrently running instances of command line adapters. If more than one instance of the Command Line Adapter 2 can be executing at the same time, you must create a dynamic unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine.</p>
Delete output file after process completes? (outputDelete)	<p>Specifies whether the output file is deleted after it is collected?</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Default ◆ No (false) <p>Note: This parameter is read-only in the GPM.</p>

GPM Configuration

The following screen shows a graphical view of some GPM parameters for the Command Line adapter. The dimmed values were specified using the Command Line adapter configuration. The active fields are env0 and env1, which cannot be configured in the service configuration.



The following example shows the corresponding business process solution using BPML.

```
<process name="Example_CommandLine2BP">
  <operation name="Command Line 2 Adapter Run Script">
    <participant name="Sample_CommandLine2_Adapter"/>
    <output message="CmdLine2InputMessage">
      <assign to="."> from="*" />
      <assign to="parm0">VAR1</assign>
      <assign to="parm1">USER</assign>
      <assign to="parm2">10</assign>
      <assign to="env0">VAR1=TEST</assign>
      <assign to="env1">USER=ME</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

The following table describes the fields used to configure the Command Line adapter in the GPM. This table contains the fields that are only configured in the GPM. Other fields may also be configured if they were left blank in the Application configuration.

Field	Description
Config (participant name)	Name of the adapter configuration. Required.
env0	An environment variable in the form name=value. Optional. Any value is valid.
env1	An environment variable in the form name=value. Optional. Any value is valid.
env2	An environment variable in the form name=value. Optional. Any value is valid.
env3	An environment variable in the form name=value. Optional. Any value is valid.
env4	An environment variable in the form name=value. Optional. Any value is valid.
env5	An environment variable in the form name=value. Optional. Any value is valid.
env6	An environment variable in the form name=value. Optional. Any value is valid.
env7	An environment variable in the form name=value. Optional. Any value is valid.
env8	An environment variable in the form name=value. Optional. Any value is valid.
env9	An environment variable in the form name=value. Optional. Any value is valid.
keepPath	Normally, any path information is stripped off the filename to allow for platform independence. This parameter allows you to keep the entire path. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes – Path information is retained ◆ No – Path information is stripped off
parm0	Resolves the \$0 placeholder. Optional. Any value is valid.
parm1	Resolves the \$1 placeholder. Optional. Any value is valid.
parm2	Resolves the \$2 placeholder. Optional. Any value is valid.
parm3	Resolves the \$3 placeholder. Optional. Any value is valid.
parm4	Resolves the \$4 placeholder. Optional. Any value is valid.
parm5	Resolves the \$5 placeholder. Optional. Any value is valid.
parm6	Resolves the \$6 placeholder. Optional. Any value is valid.
parm7	Resolves the \$7 placeholder. Optional. Any value is valid.
parm8	Resolves the \$8 placeholder. Optional. Any value is valid.
parm9	Resolves the \$9 placeholder. Optional. Any value is valid.

Field	Description
setSoTimeout	<p>Specifies, in milliseconds, how long the socket will wait in receive mode without receiving anything before timing out. This is necessary to ensure that a process doesn't "hang" indefinitely. Optional.</p> <p>Valid value: any integer. Default is 60000 milliseconds (60 seconds).</p> <p>If your command line process is going to take longer than the default 60 seconds to process completely, then increase this value accordingly.</p>
successValue	<p>If waitOnProcess is Yes (true), then this option can be used to determine what the successful return code value is. Optional.</p> <p>Valid value is any integer. The default is 0.</p> <p>If a value is specified and does not equal the return code value of the process, the business process status is set to ERROR.</p> <p>Note: The successValue parameter is an important parameter that is often overlooked. It is used to signal Application if the command line process failed. If the returned success value does not match the returned status, the process fails. Without returning a success value from an OS script, failures are not detected and the process is assumed to have passed. This creates a failure for the business functionality that is hard to correct later. In writing OS scripts, always check the return status for each call and handle it properly. This includes returning the status values to the OS shell. Error handling in scripts can cause the script to exit before the final output file is generated. Returning from the script to Application without an output file is a critical error that is handled before the returned successValue is examined. See <i>Use the output generated by the command line process?</i> on page 125 for dealing with this issue.</p> <p>Many OS commands do not return a success value, instead they output errors to stderr or stdout. In these cases, the commands stderr and/or stdout text must be captured, filtered, and an error status returned if the command failed.</p>

Output from Adapter to Business Process

The following table contains the parameters passed from the Command Line Adapter 2 to the business process:

Parameter Name and Element Value (BPML)	Description
Document (CLA2/document)	If a file is collected in non-bootstrap mode, the document is placed in ProcessData, not as the Primary Document.
DocumentId (CLA2/documentId)	If a file is collected in non-bootstrap mode, the document identifier of the document is placed here.
ProcessExitValue (CLA2/ProcessExitValue)	Sets the process data value to the exit value of the process.
FileName (CLA2/FileName)	The name of the file, if any, that was collected as part of the output from the process that ran.

Usage Examples

This section contains an example using the Command Line Adapter 2. Examples are included using both the GPM and BPML.

Invoking the Command Line Adapter to Run a Shell Script

The following example business process illustrates using the Command Line Adapter 2 to execute a shell script that expects an input file as the first parameter, an output file as the second parameter, and three other parameters.

When this example configuration is used, a shell script called “test.sh” (which resides in the /home directory) is run.

The program requires the input filename as the first parameter, the output filename as the second parameter, and three other parameters.

Because the useInput variable is set to true and the inputName variable is blank, the name of the primary document replaces the \$Input placeholder.

Because the useOutput variable is set to true and the outputName variable is blank, the \$Output placeholder is replaced with the name of the primary document.

If the document name in the workflow context is “data.txt” in this example, the command line becomes /home/test.sh data.txt data.txt VAR1 USER 10 at run-time.

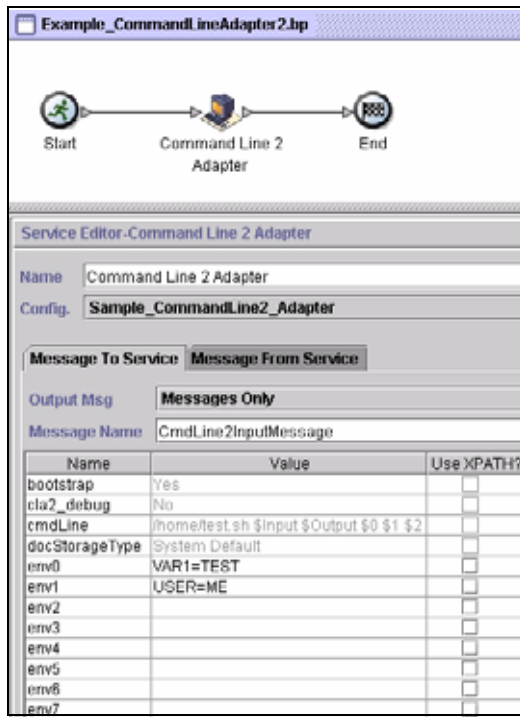
The name of the primary document is passed as the input file to the shell script program on the command line.

The name of the primary document is passed as the output file to the shell script program on the command line.

Note: If the inputName and outputName parameters had file names entered, these file names would replace the \$Input and \$Output placeholders.

GPM Example

The following example illustrates the above business process using the GPM.



Business Process Modeling Language (BPML) Example

The following example illustrates the same business process using BPML.

```
<process name="Example_CommandLine2_BP">
  <operation name="Command Line Adapter 2 Run Script">
    <participant name="Sample_CommandLine2_Adapter"/>
    <output message="CmdLine2InputMessage">
      <assign to="." from="*" />
      <assign to="parm0">VAR1</assign>
      <assign to="parm1">USER</assign>
      <assign to="parm2">10</assign>
      <assign to="env0">VAR1=TEST</assign>
      <assign to="env1">USER=ME</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

Invoking the Command Line Adapter 2

Adapter startup at the time of Application startup is driven by (Windows) startWindowsService.cmd or (UNIX and iSeries) the run.sh script.

To start the Command Line Adapter 2:

1. Locate the client .jar file (CLA2Client.jar) that contains all the necessary classes. This is located in the *install_dir/client/cmdline2* directory.
2. If the adapter is going to run on a different machine, copy the CLA2Client.jar file to the system that will run the remote adapter.

Note: The CLA2Client.jar does not need to be moved if the remote version is on the same machine.

3. To start the remote adapter, enter:

```
java -jar CLA2Client.jar < >  
Example: java -jar CLA2Client.jar 15699
```

The *port* can be any port that is not in use. This is the value to assign in the Remote Port field in the service configuration. See *Remote Port (remotePort)*. If you upgrade Application at any time, you will also need to use the corresponding new CLA2Client.jar file to avoid a “ClassConflict” error.

Note: You may also run CLA2Client.jar with these commands:

- ◆ (UNIX or iSeries) ./startCmdLine2.sh
- ◆ (Windows service) startCLA2WindowsService.cmd

Stopping the Command Line Adapter 2

If Application is shut down with the (Windows) stopWindowsService.cmd or (UNIX and iSeries) hardstop.sh script, the Command Line Adapter 2 also shuts down. You can also stop the Command Line Adapter 2 with these commands:

```
(UNIX or iSeries) ./stopCmdLine2.sh  
(Windows service) stopCLA2WindowsService.cmd
```

Otherwise, once started, the adapter runs silently as configured and does not return to the command line until it is finished, interrupted, or fails. Therefore, you cannot use that command line to execute any other commands.

Compression Service

The following table provides an overview of the Compression service:

System name	CompressionService
Graphical Process Modeler (GPM) category	All Services
Description	<p>Provides the ability to compress (deflate) a document or group of documents in the business process context. It also provides the ability to decompress (inflate) a document.</p> <p>The following options are available with the Compression service:</p> <ul style="list-style-type: none">◆ Compress all documents or just the primary document in a business process context.◆ Decompress a .zip file and put a specified document into the primary document or decompress the file and start a business process for each document. <p>With the Compress option, parameters can be set to specify whether the files should be deflated or just stored. If the files are being compressed, parameters can be set to determine the level of compression. The decompression option can start a business process for each file, or it can put a specified document in the primary document.</p>
Business usage	Used within a business process to perform compression or decompression of a document.
Usage example	If you have a zip file in the primary document, you can use this service to decompress the file and start a business process for each document found.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	If decompressing files, and Decompress_result is set to Start_bpml, child business processes are started for each document in compressed file.
Invocation	Runs as part of a business process.
Business process context considerations	This service uses the primary document and can also compress other documents in the business process context.
Returned status values	<ul style="list-style-type: none">◆ Success – Compression or decompression was successful.◆ Error – Errors were encountered during compression or decompression. The report contained in the Workflow Context Status report should be consulted for further detail.

Restrictions	When decompressing, if you choose to put a specific document in the primary document area, but fail to specify a filename when multiple files exist in the compressed input file, Application selects the first of the files present, as determined by the compression utility that produced the compressed file.
--------------	---

How the Compression Service Works

The Compression service provides two options:

Compression

Decompression

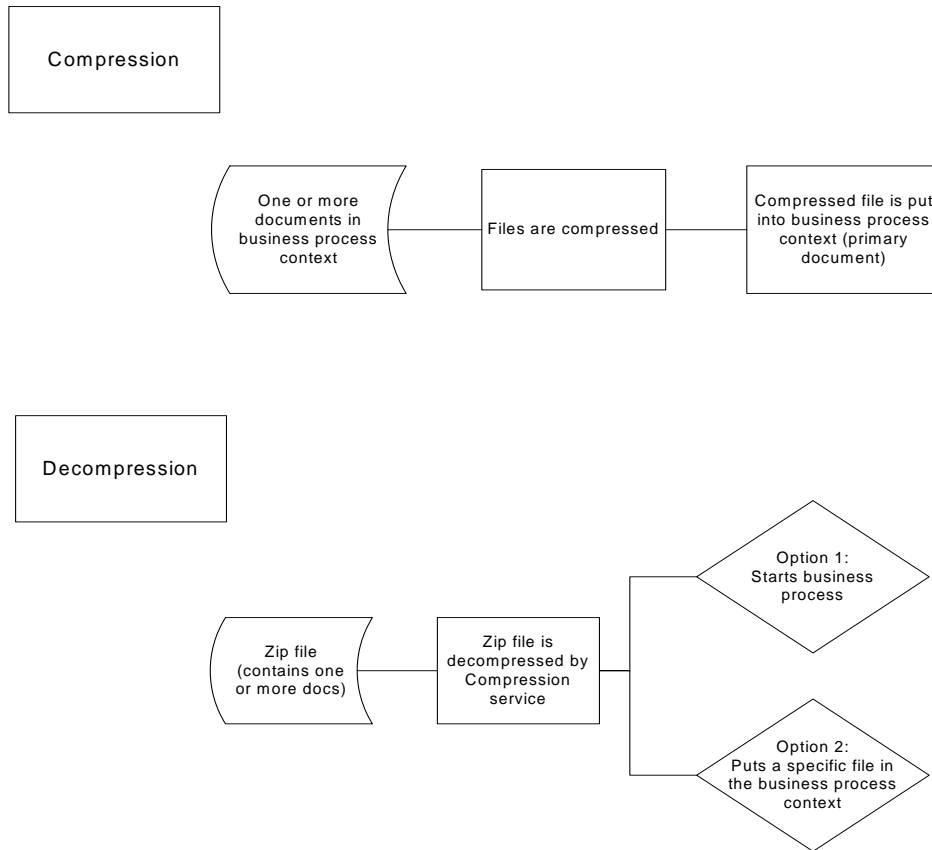
The Compression option is used to deflate files, and it enables you to define the level of compression used. You can also use the Compression option to store files without compressing them.

The Decompression option inflates a compressed file, then enables you to use the resulting decompressed files in business processes. The Decompression option can be configured to start a business process for each file, or it can put a specified document in the primary document. For example, if the primary document in a business process contained a .zip file, you could use the Compression service to decompress the file and start a business process for each document contained in the .zip file.

Note: When decompressing, if you choose to put a specific document in the primary document area, but fail to specify a filename when multiple files exist in the compressed input file, Application selects the first of the files present, as determined by the compression utility that produced the compressed file.

If the compression or decompression completed without error, a Done message is written to the status report. If it was unsuccessful, an error message is written to the status report.

The following figure shows the basic flows for compression and decompression:



Business Process Example

The business process example in this section illustrates this sequence of actions:

1. The File System adapter collects one document.
2. The business process moves the document to the business process context.
3. The File System adapter collects a second document.
4. The Compression service deflates both documents.

```
<process name="Comp_Deflate_lvl0">
  <sequence>
    <operation name="FileSystem">
      <participant name="FileSystem"/>
      <output message="outputMessage">
        <assign to="Action">FS_COLLECT</assign>
      <assign to="collectionFolder">/input</assign>
      <assign to="filter">Excel.xls</assign>
      <assign to="useSubFolders">>false</assign>
      <assign to="bootstrap">>false</assign>
      <assign to="deleteAfterCollect">>false</assign>
      <assign to="extractionFolder">//dummy</assign>
      <assign to="." from="*"></assign>
    </operation>
  </sequence>
</process>
```

```

        </output>
        <input message="inputMessage">
            <assign to="." from="*"></assign>
        </input>
    </operation>
    <assign to="ExcelDoc/@SCIOBJECTID" from="string(//PrimaryDocument/@SCIOBJECTID) "
append="true"/>
    <operation name="FileSystem">
        <participant name="FileSystem"/>
        <output message="outputMessage">
            <assign to="Action">FS_COLLECT</assign>
        <assign to ="collectionFolder"/></input</assign>
        <assign to ="filter">Text.txt</assign>
        <assign to ="bootstrap">false</assign>
        <assign to ="deleteAfterCollect">false</assign>
        <assign to ="useSubFolders">false</assign>
        <assign to ="extractionFolder">//dummy</assign>
            <assign to="." from="*"></assign>
        </output>
        <input message="inputMessage">
            <assign to="." from="*"></assign>
        </input>
    </operation>
    <operation name="Compress">
        <participant name="CompressionService"/>
        <output message="outputMessage">
            <assign to="." from="*"></assign>
            <assign to ="compression_action">compress</assign>
            <assign to ="compressed_filename">DeflatedFile.zip</assign>
            <assign to ="compression_level">0</assign>
            <assign to ="compression_type">Deflate</assign>
        </output>
        <input message="inputMessage">
            <assign to="." from="*"></assign>
        </input>
    </operation>
</sequence>
</process>

```

Implementing the Compression Service

To implement the Compression service, complete the following tasks:

1. Configure the pre-installed copy of the Compression service, or create a new configuration. For information, see *Configuring the Compression Service* on page 136.
2. Use the Compression service in a business process.

Configuring the Compression Service

To configure the Compression service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.
bpml_name	Name of the business process to start. Required if you selected Start_bpml for Decompress_result field. Valid value is a string. Decompression parameter only.
compressed_filename	What to name the compressed file. Valid value is a string: file name with extension. Default is CompressedFile<date/timestamp>.zip. Date/Timestamp is in the format CCYYMMDDHHMMSSMMS. Compression parameter only.
compression_action	Whether to compress or decompress documents. Valid values are Compress and Decompress.
compression_level	Level of compression. Valid values are 0 - 9 (default is 0). Compression parameter only.
compression_type	Compression method. Valid values are Deflate and Store (default is Deflate). Compression parameter only.
decompress_result	What to do with decompressed files. Valid values are Start a business process and Put a specific document in the primary document. Decompression parameter only. Required.
doc_to_compress	Whether to compress all docs or just the primary doc. Valid values are All and Primary Doc (default is All). Compression parameter only.
file_name	Name of file to put in primary document. Valid value is a string: name of file with extension. Defaults to first document in the list. Decompression, primary_doc parameter only. Note: When decompressing: if you choose to put a specific document in the primary document area, but fail to specify a filename when multiple files exist in the compressed input file, Application will select the first of the files present, as determined by the compression utility that produced the compressed file.

Connect:Direct Requester Adapter

The following table provides an overview of the Connect:Direct Requester adapter:

System Name	Connect Direct Requester Adapter
Graphical Process Modeler (GPM) categories	Not configurable in the GPM.
Description	This adapter configures the communication between Application and a remote Connect:Direct server. It acts as a Connect:Direct client.
Business usage	Use this adapter to request that a remote Connect:Direct node perform specific operations. For example, to view work that is active in the Connect:Direct server (for example, SELECT PROCESS) or view events that have already happened on the Connect:Direct server (for example, SELECT STATISTICS).
Usage example	The Connect:Direct Requester adapter is used to submit a process that initiates a sequence of steps on a remote Connect:Direct server.
Preconfigured?	Yes
Requires third party files?	The following third-party files are required: Connect:Direct Java Application Interface 1.0.02 CDJAI.jar (included with Application) Certicom SSL Library (included with Application)
Platform availability	All supported platforms
Related services	The Connect:Direct Requester adapter is tightly integrated with the Connect:Direct Submit Process service, Connect:Direct Requester Select Process service, and Connect:Direct Requester Select Statistics service.
Application requirements	This adapter must be used with a Perimeter server, which must be configured prior to setting up this adapter.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	None
Restrictions	None
Persistence level	None
Testing considerations	Restart Application after configuring and enabling the adapter to confirm that it can successfully connect to the remote Connect:Direct server using a supported Connect:Direct service.

Implementing the Connect:Direct Requester Adapter

To implement the Connect:Direct Requester adapter, complete the following tasks:

1. Create a Connect:Direct Requester adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the Connect:Direct Requester adapter. For information, see *Configuring the Connect:Direct Requester Adapter* on page 138.
3. Use the Connect:Direct Requester adapter in a business process.

Configuring the Connect:Direct Requester Adapter

To configure the Connect:Direct Requester adapter, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default)◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
Remote Connect:Direct Server Address	The IP address of the remote Connect:Direct Server to connect to. Required. Valid values are 0.0.0.0 to 255.255.255.254 (default is 127.0.0.1).
Remote Connect:Direct Server Port	The port number that the remote Connect:Direct Server is listening on for API connections. Required. Valid are any numeric values (default is 1364).
Remote Connect:Direct User ID	The user ID used to log on to the Connect:Direct server. Required
Remote Connect:Direct User Password	The password used to log on to the Connect:Direct server. Required.
Select Connect:Direct Perimeter Services Option	The Perimeter server to be used. Required. Default value is Local Node1.

Field	Description
Encryption using Secure+	<p>Whether Connect:Direct Secure+ should be active. Required. Valid values are Enabled and Disabled (default).</p> <p>If Enabled is selected for the Encryption using Secure+ parameter, then the following fields are displayed:</p> <ul style="list-style-type: none"> ◆ CA Certificates ◆ Cipher Suites ◆ SSL or TLS
CA Certificates	List of Secure+ CA certificates. Required if Select Connect: Direct Secure+ Option is enabled.
Cipher Suites	List of Secure+ Cipher suites. Required if Select Connect: Direct Secure+ Option is enabled.
SSL or TLS	Whether Secure Sockets Layer (SSL) or TLS should be used. Required if Select Connect: Direct Secure+ Option is enabled. Valid values are SSL (default) and TLS.

Connect:Direct Requester Select Process Service

The following table provides an overview of the Connect:Direct Requester Select Process service:

System name	Connect:Direct Requester Select Process Service
Graphical Process Modeler (GPM) categories	Applications > Sterling Commerce > Connect:Direct
Description	This service retrieves information about a process that is in progress on a remote Connect:Direct Server. Information can be retrieved based on query criteria such as Process name and number, SNODE, queue, status, or submitter node and user ID.
Business usage	A business user would use this service to get status information about a process submitted to a remote Connect:Direct node.
Usage example	A business process submits a Connect:Direct Process to a remote Connect:Direct Server, then uses the Select Process service to check the status of the Connect:Direct Process.
Preconfigured?	A Connect:Direct Requester adapter must be configured before using this service.
Requires third party files?	Connect:Direct Windows 4.0.00 or later, Connect:Direct UNIX 3.4.00 or later, or Connect:Direct OS/390 4.3.00 or later, available from Sterling Commerce.
Platform availability	All supported platforms
Related services	Connect:Direct Requester Submit Process Service and Connect:Direct Requester Select Statistics Service.
Application requirements	None
Initiates business processes?	No
Invocation	The business process running this service must be started by a user with permission to issue the Select Process command on the target C:D Server.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ SUCCESS Command was successfully executed. Select process returned N records ◆ ERROR Connect:Direct Requester Logon Error An error occurred while logging into the Connect:Direct node. ◆ ERROR Connect:Direct Requester Message Error An error occurred on the Connect:Direct node (command syntax error, user authorization error). ◆ ERROR Connect:Direct Requester Connection Error An error occurred with the connection to the Connect:Direct node.
Restrictions	None
Persistence level	System default
Testing considerations	Debug information for this service can be found in the C:D Interop log files.

Implementing the Connect:Direct Requester Select Process Service

To implement the Connect:Direct Requester Select Process service, complete the following tasks:

1. Create a Connect:Direct Requester Select Process service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Connect:Direct Requester Select Process service. For information, see *Configuring the Connect:Direct Requester Select Process Service* on page 141.
3. Use the Connect:Direct Requester Select Process service in a business process.

Configuring the Connect:Direct Requester Select Process Service

To configure the Connect:Direct Requester Select Process service, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
CDRequesterName	Specifies the Connect:Direct Requester adapter to use with this service. Required.

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Requester Select Process service:

Parameter	Description
CDRequesterName	Specifies the Connect:Direct Requester adapter to use with this service. Required.
ProcessName	Specifies the 1–8 character name of the process. Optional. Valid value is alphanumeric that begins with an alpha character.
ProcessNumber	Specifies the system-assigned number of the process. Optional. Valid value range is 1–99999.
Queue	<p>Specifies which queue to select processes from. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Hold ◆ Wait ◆ Timer ◆ Exec
Snode	Specifies the secondary node to be used in this process. The secondary node name is a 1–16 alphanumeric character name that is defined in the network map. Optional. Valid values are alphanumeric characters or nationals (@#\$.).

Parameter	Description
Status	<p>Selects processes for viewing according to status. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Execution - Process is running. ◆ Pending Execution - Process is selected to run and startup is in progress. ◆ Waiting Connection - Process is ready to run, but all available connections to the SNODE are in use. ◆ Waiting Start Time - Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired. ◆ Held Suspension - Operator issued a delete process request with Hold set to Yes. ◆ Timer Retry - Process error occurred and the process was moved to the Timer queue in Timer Retry status with short-term and long-term wait times beginning. ◆ Held for Call - Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in Waiting Connection status. The process is placed in the Execution queue when it is selected to run. ◆ Held Due to Error - session error or other abnormal condition occurred. ◆ Held Initially - Process was submitted with the Hold option set to Yes. ◆ Held By Operator - Change process request with Hold set to Yes was issued. ◆ Held By Retain - Process was submitted with retain after execution set to Yes or Initial.
Submitter	Specifies the node name and user ID of the user that submitted the process. Separate the node name and user ID with a comma; for example, (atlanta, user1), (atlanta, user2), (atlanta, user3). Optional.
Userid	User ID used to connect to the Connect:Direct server. Overrides the user ID specified in the Connect:Direct Requester adapter. Optional.
Password	Password used to connect to the Connect:Direct server. Overrides the password specified in the Connect:Direct Requester adapter. Optional.

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Requester Select Process service to the business process:

Parameter	Description
ByteCount	Specifies the number of bytes transferred so far during the current process step (if the process is executing). Optional.
Checkpoint	Indicates if Checkpoint is activated for this process. Optional.
Class	Determines the node-to-node session on which a process can be executed. Optional. Valid values are 1-10.

Parameter	Description
ConditionCode	<p>Specifies the return code values associated with step termination. Optional. Valid codes are:</p> <ul style="list-style-type: none"> ◆ 0 – Successful execution of the process. ◆ 4 – A warning error was encountered. The statement probably finished normally, but you should verify the execution results. ◆ 8 – An error occurred during process execution. ◆ 16 – A severe error occurred during process execution.
DestinationDisposition1	<p>Specifies what to do with the destination file after a copy is complete. Optional. The destination DISP valid values are:</p> <ul style="list-style-type: none"> ◆ NEW – Creates a new file on the destination node. ◆ RPL – Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node. ◆ MOD – Appends data to the end of an existing file for which you have exclusive rights.
DestinationDisposition2	<p>Specifies the disposition of the destination file after a successful process step termination that results in a zero completion code. Optional. The destination DISP valid values are:</p> <ul style="list-style-type: none"> ◆ NEW – Creates a new file on the destination node. ◆ RPL – Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node. ◆ MOD – Appends data to the end of an existing file for which you have exclusive rights.
DestinationDisposition3	<p>Specifies the disposition of the destination file after an abnormal process step termination that results in a non-zero completion code. Optional. The destination DISP valid values are:</p> <ul style="list-style-type: none"> ◆ NEW–Creates a new file on the destination node. ◆ RPL–Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node. ◆ MOD –Appends data to the end of an existing file for which you have exclusive rights.
DestinationFile	Specifies the destination file used in the process. Optional.
EncryptionAlgorithmName	Specifies the name of the encryption algorithm. Optional.
ExecutionPriority	<p>Priority under which the operating system thread that executes Connect:Direct runs. Optional. Valid values are 1-15.</p> <p>Note: Applies to Windows only.</p>
ExtendedCompression	Extended compression option. Optional.
Feedback	Feedback code for the module. Optional. Valid value depends on the module that creates it.
FromNode	Node that sent the file. Optional. Valid values are S–SNODE and P–PNODE.

Parameter	Description
Function	Function being performed. Optional.
Hold	<p>Holds status of a process. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ No – Do not place process in the Hold queue. It is runs as soon as resources are available. ◆ Yes – Hold process in queue in Held Initially (HI) status until it is explicitly released. ◆ Call – Hold process until the SNODE, as specified in the process SNODE parameter, connects to the PNODE. The process is then released for execution. The process is also released when another process on the PNODE connects to the SNODE.
LocalNode	Node that processed the file. Optional. Valid values are S–SNODE and P–PNODE.
LogDateTime	Specifies the date and time that the statistics record was written to the statistics log. Optional.
MessageData	Specifies the short text associated with the message ID. Optional.
MessageId	Current message ID for the process. Optional.
Pnode	Primary node. Optional.
PnodePlexClass	Pnode class that directs the process to only certain servers in a Connect:Direct/Plex environment. Optional.
Priority	Priority assigned to the process. The lower the number the higher the priority. Optional. Valid values are 1-15.
ProcessName	Process name. Optional.
ProcessNumber	Process number. Optional.
Queue	<p>Queue containing the process. Optional. Valid queues are:</p> <ul style="list-style-type: none"> ◆ All–All processes in the TCQ. ◆ Execution–Processes currently being executed. ◆ Hold–Processes that are either held by the user or operator or held due to execution errors. ◆ Timer–Processes that are scheduled to be executed later, or processes in time retry due to session errors. ◆ Wait–Processes that are eligible for execution and are awaiting selection.
RecordCount	Number of records in the file that was sent. Optional.
Restart	Whether Restart is activated for this process. Optional.

Parameter	Description
Retain	Whether Connect:Direct retains a copy of a process after it is executed. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Initial – Specifies to retain the process in the Hold queue for execution every time that Connect:Direct initializes. ◆ No – Specifies not to retain the process after it is executed. ◆ Yes – Specifies to retain the process in the Hold queue after it is executed. You can release the process for execution later or delete it.
ScheduledDateTime	Date and time that a process is scheduled to be executed. Optional.
SecureEnabled	Whether Secure+ is activated for the process. Optional.
Signature	Whether digital signature is activated. Optional.
Snode	Secondary node used for the process. Optional.
SnodePlexClass	Snode class that directs the process to only certain servers in a Connect:Direct/Plex environment. Optional.
SourceDisposition1	Access to the source file during a copy operation. Optional. Source DISP valid values are: <ul style="list-style-type: none"> ◆ SHR – The file can be opened for read-only access while it is being copied. ◆ OLD – The file cannot be opened during the transfer.
SourceDisposition2	Disposition of the source file after a successful process step termination that results in a zero completion code. Optional. Source DISP valid values are: <ul style="list-style-type: none"> ◆ SHR – The file can be opened for read-only access while it is being copied. ◆ OLD – The file cannot be opened during the transfer.
SourceDisposition3	Disposition of the source file after an abnormal process step termination that results in a non-zero completion code. Optional. Source DISP valid values are: <ul style="list-style-type: none"> ◆ SHR – The file can be opened for read-only access while it is being copied. ◆ OLD – The file cannot be opened during the transfer.
SourceFile	Source file used in the process. Optional.
StandardCompression	Standard compression option. Optional.

Parameter	Description
Status	<p>Process status. Optional. Valid statuses are:</p> <ul style="list-style-type: none"> ◆ EX (Execution) Process is executing. ◆ PE (Pending Execution) Process is selected for execution and startup is in progress. ◆ WC (Waiting Connection) Process is ready to execute, but all available connections to the SNODE are in use. ◆ WA (Waiting Start Time) Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired. ◆ HS (Held Suspension) Operator issued a delete process request with Hold set to Yes. ◆ RE (Timer Retry) Process error occurred and the process was moved to the Timer queue in RE status with short-term and long-term wait times beginning. ◆ HC (Held for Call) Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in WC status. The process is placed in the Execution queue when it is selected for execution. ◆ HE (Held Due to Error) session error or other abnormal condition occurred. ◆ HI (Held Initially) Process was submitted with the Hold option set to Yes. ◆ HO (Held By Operator) Change process request with Hold set to Yes was issued. ◆ HR (Held By Retain) Process was submitted with retain after execution set to Yes or Initial. ◆ RS (Restart) Process is restarting.
StepName	Current process step. Optional.
SubmitNode	Node from which the process was submitted. Optional.
Submitter	User ID that submitted the process. Optional.
SubmitterNode	Node from which the process was submitted. Optional.
XmitBytes	Number of bytes transmitted. Optional.
XmitRUs	Number of network buffers transmitted. Optional.

Business Process Example

The following example business process issues a Select Process command on a remote Connect:Direct server to retrieve all Connect:Direct processes on the queue and puts the output into a file:

```
<process name="CDInterop_testSelectProcess_GENERIC">
  <sequence>
    <operation name="CDSelectProcess">
      <participant name="CDRequesterSelectProcess"/>
      <output message="CDSelectProcess">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">

```

```

        <assign to="SelectProcResults" from="*"></assign>
    </input>
</operation>

<!-- Move DOM into PrimaryDocument to setup the File System Adapter -->
    <assign to="." from="DOMToDoc(/ProcessData, 'PrimaryDocument', 'no',
'ProcessData')"></assign>

<!-- Load the process id used for unique naming of the process data output file -->
    <operation name="GetProcessId">
        <participant name="This"/>
        <output message="thisRequest">
            </output>
        <input message="thisResponse">
            <assign to="ConfigData/wfId" from="//INVOKE_ID_LIST/text()" append="true"/>
        </input>
    </operation>

<!-- FSA to Extract ProcessData -->
    <operation name="File System Adapter">
        <participant name="CDInteropTestFSA"/>
        <output message="FileSystemInputMessage">
            <assign to="Action">FS_EXTRACT</assign>
            <assign to="appendOnExtract">false</assign>
            <assign to="assignedFilename" from="concat(ConfigData/wfId/text(),
'_ProcessDataOut.xml')"/>
            <assign to="assignFilename">true</assign>
            <assign to="bootstrap">false</assign>
            <assign to="extractionFolder">/usr/home/out</assign>
            <assign to="useSubFolders">false</assign>
            <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
            <assign to="." from="*"></assign>
        </input>
    </operation>

    <onFault>
        <sequence name="OnFault">
<!-- Move DOM into PrimaryDocument to setup the File System Adapter -->
            <assign to="." from="DOMToDoc(/ProcessData, 'PrimaryDocument', 'no',
'ProcessData')"></assign>

<!-- Load the process id used for unique naming of the process data output file -->
            <operation name="GetProcessId">
                <participant name="This"/>
                <output message="thisRequest">
                    </output>
                <input message="thisResponse">
                    <assign to="ConfigData/wfId" from="//INVOKE_ID_LIST/text()" append="true"/>
                </input>
            </operation>

<!-- FSA to Extract ProcessData -->
            <operation name="File System Adapter">
                <participant name="CDInteropTestFSA"/>

```

```
<output message="FileSystemInputMessage">
  <assign to="Action">FS_EXTRACT</assign>
  <assign to="appendOnExtract">false</assign>
  <assign to="assignedFilename" from="concat(ConfigData/wfId/text(),
'_ProcessDataOut.xml')"/>
  <assign to="assignFilename">true</assign>
  <assign to="bootstrap">false</assign>
  <assign to="extractionFolder">/usr/home/out</assign>
  <assign to="useSubFolders">false</assign>
  <assign to="." from="*"></assign>
</output>
<input message="inmsg">
  <assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</onFault>
</sequence>
</process>
```

Connect:Direct Requester Select Statistics Service

The following table provides an overview of the Connect:Direct Requester Select Statistics service:

System Name	Connect:Direct Requester Select Statistics Service
Graphical Process Modeler (GPM) categories	Applications > Sterling Commerce > Connect:Direct
Description	Retrieves statistics from a remote Connect:Direct node based on various criteria, including process name, start and stop date, start and stop time. This service examines records in the Statistics log for a remote Connect:Direct node.
Business usage	A business user would use this service to get status information about a process submitted to a remote Connect:Direct node.
Usage example	A business process submits a Connect:Direct process to a remote Connect:Direct server, then uses the Select Statistics service to check the completion status of the Connect:Direct process.
Preconfigured?	A Connect:Direct Requester adapter must be configured before using this service.
Requires third party files?	Connect:Direct Windows 4.0.00 or later, Connect:Direct UNIX 3.4.00 or later, or Connect:Direct OS/390 4.3.00 or later, available from Sterling Commerce.
Platform availability	All supported platforms
Related services	Connect:Direct Requester Submit Process service and Connect:Direct Requester Select Process service.
Application requirements	None
Initiates business processes?	No
Invocation	The business process running this service must be started by a user with permission to issue the Select Process command on the target Connect:Direct server.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ SUCCESS Command was successfully executed. Select statistics returned N records ◆ ERROR Connect:Direct Requester Logon Error An error occurred while logging into the Connect:Direct node. ◆ ERROR Connect:Direct Requester Message Error An error occurred on the Connect:Direct node (command syntax error, user authorization error). ◆ ERROR Connect:Direct Requester Connection Error An error occurred with the connection to the Connect:Direct node.
Restrictions	None
Persistence level	System default
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Requester Select Statistics Service

To implement the Connect:Direct Requester Select Statistics service, complete the following tasks:

1. Create a Connect:Direct Requester Select Statistics service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Connect:Direct Requester Select Statistics service. For information, see *Configuring the Connect:Direct Requester Select Statistics Service* on page 151.
3. Use the Connect:Direct Requester Select Statistics service in a business process.

Configuring the Connect:Direct Requester Select Statistics Service

To configure the Connect:Direct Requester Select Statistics service, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
CDRequesterName	Specifies the Connect:Direct Requester adapter to use with this service. Required.

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Requester Select Statistics service:

Parameter	Description
CDRequesterName	Specifies the Connect:Direct Requester adapter to use with this service. Required.
ConditionCodeOperation	<p>Comparison operator used to select statistics based on the return code value associated with step termination. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Equal ◆ Greater Than ◆ Greater Than or Equal To ◆ Less Than ◆ Less Than or Equal To ◆ Not Equal To
ConditionCodeValue	<p>Value of the return code associated with step termination. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 0 – Successful execution of the process. ◆ 4 – A warning error was encountered. The statement probably finished normally, but you should verify the execution results. ◆ 8 – An error occurred during process execution. ◆ 16 – A severe error occurred during process execution.
DestinationFile	Destination file used in the process. Optional.
ProcessNumber	System-assigned number of the process. Optional. Valid value range is 1–99999.
ProcessName	The 1–8 character name of the process. Optional. Valid value is alphanumeric that begins with an alpha character.

Parameter	Description
RecordCategory	Which record category to select statistics from. Optional. Valid values are: <ul style="list-style-type: none"> ◆ CAPR – Process records ◆ CAEV – Event records
RecordId	Which record ID to select statistics from. Optional.
Snode	Secondary node to be used in this process. The secondary node name is a 1–16 alphanumeric character name that is defined in the network map. Optional.
SourceFile	Source file used in the process. Optional.
StartDate	Day or date to start selecting statistics records. Optional.
StartTime	Time to start selecting statistics records. Format is HH:MM [XM]. Optional.
StopDate	Day or date to stop selecting statistics records. Optional.
StopTime	Time to stop selecting statistics records. Format is HH:MM [XM]. Optional.
Submitter	User ID that submitted the process. Optional.
UserId	User ID used to connect to the Connect:Direct server. Overrides the user ID specified in the Connect:Direct Requester adapter. Optional.
Password	Password used to connect to the Connect:Direct server. Overrides the password specified in the Connect:Direct Requester adapter. Optional.

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Requester Select Statistics service to the business process:

Parameter	Description
AliasMember	Alias name of the member copied. Optional.
BytesRead	Number of bytes read from the source file. Optional.
BytesReceived	Number of bytes received by the destination file. Optional.
BytesSent	Number of bytes sent to the destination file. Optional.
BytesWritten	Number of bytes written to the destination file. Optional.
Checkpoint	Whether checkpointing is activated for this process. Optional.
Class	Node-to-node session on which a process can be executed. Optional.

Parameter	Description
ConditionCode	Return code values for the step termination. Optional. Valid values are: <ul style="list-style-type: none"> ◆ 0 – Successful process execution. ◆ 4 – A warning error was encountered. The statement probably completed normally, but you should verify the execution results. ◆ 8 – An error occurred during process execution. ◆ 16 – A severe error occurred during process execution.
ControlBlockEncryptionAlgorithm	Algorithm used to encrypt Connect:Direct control blocks used for strong authentication. This is the first algorithm ID in the PNODE list that is also in the SNODE list. Optional.
CurrentSignatureVerified	Whether the current encryption key was used for verifying the digital signature. Optional.
DestinationDisposition1	What to do with the destination file after a copy is complete. Optional. Valid values are: <ul style="list-style-type: none"> ◆ NEW – Creates a new file on the destination node. ◆ RPL – Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node. ◆ MOD – Appends data to the end of an existing file for which you have exclusive rights.
DestinationDisposition2	Disposition of the destination file after a successful process step termination that results in a zero completion code. Optional.
DestinationDisposition3	Disposition of the destination file after an abnormal process step termination that results in a non-zero completion code. Optional.
DestinationFile	Name of the destination file. Optional.
ExecutionPriority	Priority under which the operating system thread that executes Connect:Direct runs. Optional. Note: Applies to Windows only.
ExtendedCompression	Extended compression option. Optional.
Feedback	Feedback code for the module. Applies to Windows only. Valid values depend on the module that creates it.
FromNode	Node that sent the file. Optional. Valid values are S–SNODE and P–PNODE.
Function	Function being performed. Optional.

Parameter	Description
Hold	Hold status of a process. Optional. Valid values are: <ul style="list-style-type: none"> ◆ No – The process is not placed in the Hold queue. It is executed as soon as resources are available. ◆ Yes – The process is held in the Hold queue in Held Initially (HI) status until it is explicitly released. ◆ Call – The process is held until the SNODE, as specified in the process SNODE parameter, connects to the PNODE. The process is then released for execution. The process is also released when another process on the PNODE connects to the SNODE.
LinkFail	Whether a link failure occurred during transmission. Optional.
LocalConditionCode	Condition code produced by the local node. Optional.
LocalMessageId	Message ID produced by the local node. Optional.
LocalNode	Node that processed the file. Optional. Valid values are S–SNODE and P–PNODE.
LogDateTime	Date and time that the statistics record was written to the statistics log. Optional.
MemberName	Name of the member copied. Optional.
MergeEncryptionAlgorithm	Merged data encryption algorithm resulting from the merger of the PNODE and SNODE encryption algorithms. Optional.
MergeSignature	Merged results from the digital signature settings for the PNODE and SNODE. If digital signatures are enabled for either the PNODE or the SNODE, then digital signatures are used for the session. If digital signatures are not enabled for both the PNODE and SNODE, digital signatures are not used. Optional.
MessageId	Current message for the process. Optional.
MessageShortText	Short text for the message ID. Optional.
OtherConditionCode	Condition code produced by the other (remote) node. Optional.
OtherMessageId	Message ID produced by the other (remote) node. Optional.
Pnode	Primary node. Optional.
PnodeAccounting	User-entered PNODE accounting information. Optional.
PnodeEncryptionAlgorithmList	Data encryption algorithm used on the PNODE. Optional.
PnodeEncryptionData	PNODE encryption data. Optional.
PnodePlexClass	PNODE Connect:Direct/Plex class. Optional.
PnodeSignature	Whether digital signatures are enabled for the PNODE. Optional.
PreviousSignatureVerified	Whether the previous encryption key was used for verifying the digital signature. Optional.
Priority	Priority assigned to the process. The lower the number the higher the priority. Optional.

Parameter	Description
ProcessName	Process name. Optional.
ProcessNumber	Process number. Optional.
Queue	Queue containing the process. Optional.
RecordCategory	Whether the record is related to an event or a process. Optional. Valid values are: <ul style="list-style-type: none"> ◆ CAEV – The record is related to a Connect:Direct event, such as a Connect:Direct shutdown. ◆ CAPR – The record is related to a Connect:Direct process.
RecordId	Record type indicator. Optional.
RecordsRead	Number of records read from the source file. Optional.
RecordsWritten	Number of records written to the destination file. Optional.
Restart	Whether the process was restarted. Optional.
Retain	Whether Connect:Direct retains a copy of a process after it is executed. Optional.
RUSize	Size of buffers received by the destination file. Optional.
RUsReceived	Number of buffers received by the destination file. Optional.
RUsSent	Number of buffers sent to the destination file. Optional.
ScheduledDateTime	Date and time that a process is scheduled to be executed. Optional.
SecurePlusEnabled	Indicates that Secure+ is activated for the process. Optional.
ServerName	Connect:Direct node name. Optional.
Snode	Name of the secondary node. Optional.
SnodeAccounting	User-entered SNODE accounting information. Optional.
SnodeEncryptionAlgorithmList	Data encryption algorithm used on the SNODE. Optional.
SnodeEncryptionData	SNODE encryption data. Optional.
SnodePlexClass	SNODE Connect:Direct/Plex class. Optional.
SnodeSignature	Whether digital signatures are enabled for the SNODE. Optional.
SourceDisposition1	Access to the source file during a copy operation. Optional.
SourceDisposition2	Disposition of the source file after following a successful process step termination that results in a zero completion code. Optional.
SourceDisposition3	Disposition of the source file after following an abnormal process step termination that results in a non-zero completion code. Optional.
SourceFile	Name and location of the source file. Optional.
SourceMember	Source name of the member copied. Optional.
StandardCompression	Specifies the standard compression option. Optional.

Parameter	Description
StartDateTime	Date and time that the process started. Optional.
Status	<p>Process status. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Execution - Process is executing. ◆ Pending Execution - Process is selected for execution and startup is in progress. ◆ Waiting Connection - Process is ready to execute, but all available connections to the SNODE are in use. ◆ Waiting Start Time - Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired. ◆ Held Suspension - Operator issued a delete process request with Hold set to Yes. ◆ Timer Retry - Process error occurred and the process was moved to the Timer queue in Timer Retry status with short-term and long-term wait times beginning. ◆ Held for Call - Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in Waiting Connection status. The process is placed in the Execution queue when it is selected for execution. ◆ Held Due to Error - Session error or other abnormal condition occurred. ◆ Held Initially - Process was submitted with the Hold option set to Yes. ◆ Held By Operator - Change process request with Hold set to Yes was issued. ◆ Held By Retain - Process was submitted with retain after execution set to Yes or Initial. ◆ Restart - Process is restarting.
StepName	Process step. Optional.
StopDateTime	Date and time that the process stopped. Optional.
SubmitDateTime	Date and time that the process was submitted. Optional.
SubmitNode	Node from which the process was submitted. Optional.
Submitter	User ID that submitted the process. Optional.
SubmitterNode	Node from which the process was submitted. Optional.
Sysopts	Platform-specific system operations. These parameters specify the data type, translation tables, inherited rights, attributes, and trustees. Optional.
TargetMember	Target name of the member copied. Optional.
Translation	Whether the data was translated. Optional.

Business Process Example

The following example business process issues a select statistics command to find all statistics records that have an associated process name of TEST.

```
<process name="CDSelStat">
  <sequence>
    <operation name="CDSelStat">
      <participant name="CDSelStat"/>
      <output message="CDSelStat">
        <assign to="CDRequesterName">CDReq</assign>
        <assign to="." from="*"></assign>
        <assign to="ProcessName">TEXT</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Connect:Direct Requester Submit Process Service

The following table provides an overview of the Connect:Direct Requester Submit Process service:

System Name	Connect:Direct Requester Submit Process Service
Graphical Process Modeler (GPM) categories	Applications > Sterling Commerce > Connect:Direct
Description	This service is used to submit a Connect:Direct process to a remote Connect:Direct node.
Business usage	A business user would use this service to initiate work on remote Connect:Direct node.
Usage example	A business user has a file that exists on a remote node where Connect:Direct is running. This file is needed on another node. The user creates a business process that submits a Connect:Direct Process to a remote Connect:Direct Server to transfer the file from one node to the other.
Preconfigured?	A Connect:Direct Requester Adapter must be configured before using this service.
Requires third party files?	Connect:Direct Windows 4.0.00 or later, Connect:Direct UNIX 3.4.00 or later, or Connect:Direct OS/390 4.3.00 or later, available from Sterling Commerce.
Platform availability	All supported platforms
Related services	Related services include Connect:Direct Requester Select Process Service and Connect:Direct Requester Select Statistics Service.
Application requirements	None
Initiates business processes?	No
Invocation	The business process running this service must be started by a user with permission to issue the Submit command on the target Connect:Direct Server.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ SUCCESS – Command was successfully executed.◆ ERROR – An error occurred while processing the command.
Restrictions	None
Persistence level	System default
Testing considerations	Debug information for this service can be found in the C:D Interop log files.

Implementing the Connect:Direct Requester Submit Process Service

To implement the Connect:Direct Requester Submit Process service, complete the following tasks:

1. Create a Connect:Direct Requester Submit Process service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Connect:Direct Requester Submit Process service. For information, see *Configuring the Connect:Direct Requester Submit Process Service* on page 160.
3. Use the Connect:Direct Requester Submit Process service in a business process.

Configuring the Connect:Direct Requester Submit Process Service

To configure the Connect:Direct Requester Submit Process service, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
CDRequesterName	Specifies the Connect:Direct Requester adapter to use with this service. Required.

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Requester Submit Process service:

Parameter	Description
CDRequesterName	Specifies the Connect:Direct Requester adapter to use with this service. Required.
Class	Specifies the preferred session class for the process. The process can execute in the class specified or any higher class. Optional. This value overrides any defaults. Valid values range from one to the maximum number of PNODE sessions defined in the network map definition (default is 1).

Parameter	Description
Hold	<p>Specifies the Hold status of a process. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ No – Do not place process in the Hold queue. It is executed as soon as resources are available (default). ◆ Yes – Hold process in the Hold queue in Held Initially (HI) status until it is explicitly released. ◆ Call – Hold process until the SNODE, as specified in the process SNODE parameter, connects to the PNODE. The process is then released for execution. The process is also released when another process on the PNODE connects to the SNODE.
MaxDelay	<p>Specifies the time to wait until a process has executed. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Unlimited – Wait indefinitely or until the process is complete. ◆ Queued – Wait until the process is submitted and placed on the queue. ◆ hh:mm:ss – Wait for a specific length of time. ◆ 0 – The process must be placed on the Exec queue immediately or the submission of the process is considered unsuccessful.
NewName	<p>Specifies the new name of the process. The default is the label on the process statement. Optional.</p>
NotifyUser	<p>Specifies the user ID to receive process completion messages. The user ID is notified through a Microsoft Exchange E-mail, a Windows dialog box, or a TSO notify. Optional.</p>
PnodeAccounting	<p>Specifies user data passed to the PNODE. Optional.</p>
PnodePassword	<p>Specifies the user password on the PNODE. This field is case-sensitive. Optional.</p>
PnodeUserid	<p>Specifies the user ID used as a security ID on the PNODE. This ID must be the name of an existing user account. This field is case-sensitive. Optional.</p>
Priority	<p>Specifies the priority of a process in the Transmission Control queue. Connect:Direct uses the Priority parameter for process selection. The lower the number, the higher the priority. A process with higher priority is selected for execution before a process with a lower priority. This parameter does not affect the priority during transmission. Optional. Valid values are 1–15 (default is 7).</p>
ProcessDocumentId	<p>Specifies the Document ID of the document that contains text of the process to be submitted to the Connect:Direct server.</p> <p>Optional (if both ProcessText and ProcessDocumentId are blank, then the primary document must contain the process text).</p>
ProcessText	<p>Specifies the text of the process to be submitted to the Connect:Direct server.</p> <p>Optional (If both ProcessText and ProcessDocumentId are blank, then the primary document must contain the process text).</p>

Parameter	Description
Retain	<p>Whether Connect:Direct retains a copy of a process after it is runs. If you specify Retain with a start time, the process is released to run at the specified time. Each time a retained process is released, Connect:Direct creates a copy with a new process number. The copy is executed, and the original process remains in the queue. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Initial – Retain the process in the Hold queue for execution every time that Connect:Direct initializes. Do not specify a start time if you choose this option. ◆ No – Do not retain the process after it is executed (default). ◆ Yes – Retain the process in the Hold queue after it is executed. You can release the process for execution later or delete it. When you specify a start date and start time, set Retain to Yes to continually execute the process at the scheduled time.
Snode	Specifies the secondary node to be used in this process. The secondary node name is a 1–16 alphanumeric character name that is defined in the network map. Optional. Valid values are alphanumeric or nationals (@#\$.).
SnodeAccounting	Specifies user data passed to the SNODE. Optional.
SnodePassword	Specifies the user password on the SNODE. This field is case-sensitive. Optional.
SnodeUserid	Specifies the user ID used as a security ID on the SNODE. This field is case-sensitive. Optional.
StartDate	Specifies the day or date to execute the process. Optional.
StartTime	Specifies the time to execute the process. Format is <i>HH:MM [XM]</i> . Optional.
Variables	<p>Specifies the symbolic variables to be used in the process. Symbolic variables are text strings in a process that are replaced with predefined values when the process is executed. This allows you to easily change processes when these values change.</p> <p>For example, you can define a variable <i>&filename</i> and declare the value to be file1.txt. Whenever the process encounters the <i>&filename</i> variable, it substitutes file1.txt. If you want to use a different value, just change the <i>&filename</i> definition to the new file name, and the process will use that instead. Optional.</p>
Userid	User ID used to connect to the Connect:Direct server. Overrides the user ID specified in the Connect:Direct Requester adapter. Optional.
Password	Password used to connect to the Connect:Direct server. Overrides the password specified in the Connect:Direct Requester adapter. Alphanumeric. Optional.

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Requester Submit Process service to the business process:

Parameter	Description
Class	Determines the node-to-node session on which a process can be executed. Optional. Valid values are 1–10.

Parameter	Description
ConditionCode	Specifies the return code values associated with step termination. Optional. Valid values are: <ul style="list-style-type: none"> ◆ 0 – Successful execution of the process. ◆ 4 – A warning error was encountered. The statement probably finished normally, but you should verify the execution results. ◆ 8 – An error occurred during process execution. ◆ 16 – A severe error occurred during process execution.
ExecutionPriority	The priority under which the operating system thread that executes Connect:Direct runs. Valid values are 1–15. Optional. Note: Applies to Windows only.
LogDateTime	Specifies the date and time that the statistics record was written to the statistics log. Optional.
MessageId	Specifies the current message ID for the process. Optional.
MessageText	Specifies the short text associated with the message ID. Optional.
Pnode	Specifies the primary node. Optional.
Priority	Specifies the priority assigned to the process. Optional. The lower the number the higher the priority. Valid values are 1–15.
ProcessName	Specifies the process name. Optional.
ProcessNumber	Specifies the process number. Optional.
Queue	Specifies the queue containing the process. Optional. Valid queue values are: <ul style="list-style-type: none"> ◆ All–All processes in the TCQ. ◆ Execution–Processes currently being executed. ◆ Hold–Processes that are either held by the user or operator or held due to execution errors. ◆ Timer–Processes that are scheduled to be executed later, or processes in time retry due to session errors. ◆ Wait–Processes that are eligible for execution and are awaiting selection.
Snode	Specifies the secondary node used for the process.

Parameter	Description
Status	<p>Process status. Optional. Valid status values are:</p> <ul style="list-style-type: none"> ◆ Execution -Process is executing. ◆ Pending Execution -Process is selected for execution and startup is in progress. ◆ Waiting Connection -Process is ready to execute, but all available connections to the SNODE are in use. ◆ Waiting Start Time -Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired. ◆ Held Suspension - Operator issued a delete process request with Hold set to Yes. ◆ Timer Retry - Process error occurred and the process was moved to the Timer queue in Timer Retry status with short-term and long-term wait times beginning. ◆ Held for Call - Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in Waiting Connection status. The process is placed in the Execution queue when it is selected for execution. ◆ Held Due to Error - Session error or other abnormal condition occurred. ◆ Held Initially - Process was submitted with the Hold option set to Yes. ◆ Held By Operator - Change process request with Hold set to Yes was issued. ◆ Held By Retain - Process was submitted with Retain After Execution set to Yes or Initial.
Submitter	Specifies the user ID that submitted the process. Optional.
SubmitterNode	Specifies the node from which the process was submitted. Optional.

Business Process Example

The following example business process illustrates submitting a process using text that is specified in the ProcessText field within the business process:

```

<process name="CDSubProc">
  <sequence>
    <operation name="CDSubProc">
      <participant name="CDSubProc" />
      <output message="CDSubProc">
        <assign to="CDRequesterName">CDReq</assign>
        <assign to="." from="*"></assign>
        <assign to="ProcessText">PULL PROCESS
          SNODE=cdnodename HOLD=yes
STEP02
COPY
FROM (
  FILE="/usr/test_data/inputfile"
  SNODE)
TO (
  FILE="/usr/test_data/outputfile "
  PNODE
  DISP=RPL)
PEND</assign>
</output>
<input message="inmsg">

```

```
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>
```

Connect:Direct Server Adapter (Build 4300 - Build 4308)

The following table provides an overview of the Connect:Direct Server adapter:

System Name	Connect:Direct Server Adapter
Graphical Process Modeler (GPM) category	Not configurable in the GPM.
Description	This adapter receives and processes requests from remote Connect:Direct nodes.
Business usage	Use this adapter to: <ul style="list-style-type: none">◆ Mediate requests originating from a business process and going to an external trading partner using a (non-Gentran Integration Suite) Connect:Direct node or another Connect:Direct Server adapter.◆ Accept requests from a Connect:Direct node (or another Connect:Direct Server adapter) to copy to or from a mailbox or business process.◆ Forward requests from one remote Connect:Direct node to another
Usage example	A trading partner submits a Connect:Direct process on their local system. The process that runs must initiate a session with the Gentran Integration Suite Connect:Direct node and copy a file. The Connect:Direct Server adapter allows the session and responds to the copy request.
Preconfigured?	No preconfigured copy of the adapter is provided.
Requires third party files?	The following files external to Gentran Integration Suite are required: <ul style="list-style-type: none">◆ Connect:Direct Server◆ Certicom SSL Library (included with Gentran Integration Suite)◆ JZLIB (included with Gentran Integration Suite)◆ CDJava.jar (build 81) (included with Gentran Integration Suite)
Platform availability	All supported Gentran Integration Suite platforms

Related services	<p>To use the Connect:Direct Server adapter as a PNODE, you must configure and use one or more of the following services:</p> <ul style="list-style-type: none"> ◆ Connect:Direct Server Begin Session Service ◆ Connect:Direct Server BP Response Service ◆ Connect:Direct Server CopyFrom Service ◆ Connect:Direct Server CopyTo Service ◆ Connect:Direct Server End Session Service ◆ Connect:Direct Server Run Job Service ◆ Connect:Direct Server Run Task Service ◆ Connect:Direct Server Submit Service <p>The above services can be used as part of a business process. When invoked, they use a Connect:Direct Server adapter to perform specified activities with a remote Connect:Direct node. The adapter and remote node context are provided by a session token created by the Connect:Direct Server Begin Session Service. Subsequent services in the business process refer to this session token.</p>
Application requirements	<p>The following are optional choices for interoperability with Connect:Direct.</p> <ul style="list-style-type: none"> ◆ Connect:Direct Windows 4.1.00 (with patch 26) or later ◆ Connect:Direct UNIX 3.6.00 or later ◆ Connect:Direct OS/390 4.4.00 or later ◆ Connect:Direct HP Non-Stop ◆ Connect:Direct Select
Initiates business processes?	Initiates the business process specified by the remote Connect:Direct node or a business process selected as a target for a copy.
Invocation	The adapter is started when Gentran Integration Suite is started.
Business process context considerations	None
Returned status values	None

Restrictions	<p>The Connect:Direct Server adapter includes parameters to govern checkpoint/restart behavior if either Gentran Integration Suite or the remote node fails. If the remote Connect:Direct Server or the network fails during a copy operation, the adapter goes into a retry mode using the values specified at configuration. The adapter waits the configured amount of time and then resumes the copy.</p> <p>The length of time the adapter waits to resume the copy is impacted by two values:</p> <ul style="list-style-type: none"> ◆ Interval between Retry attempts (seconds), as specified in the adapter configuration. Default is 1 minute. ◆ Frequency defined by the scheduled system business process BPEXpirator. BPEXpirator is used in the Wait service. The Wait service is used when a system failure interrupts a Connect:Direct copy operation. Default is 15 minutes. <p>Whichever value is longer governs behavior. By coordinating the adapter retry interval and the BPEXpirator schedule, you can configure a window of time appropriate for your needs.</p> <p>Gentran Integration Suite enables automatic restart when the recovery level is set to Auto-Resume. Checkpoint restart is only enabled for file system-based document, outbound (from Gentran Integration Suite) data transfers. (If Gentran Integration Suite is the Pnode this means business processes using the CopyTo Service. If Gentran Integration Suite is the Snode this means the remote Connect:Direct Server copies documents from Gentran Integration Suite to the remote node). All other restarted copy operations are started from the beginning of the document.</p>
Persistence level	None
Testing considerations	Debug information for this adapter can be found in the Connect:Direct Server adapter log files.

Implementing the Connect:Direct Server Adapter

To implement the Connect:Direct Server adapter, complete the following tasks:

1. Collect information about any remote Connect:Direct Servers that will communicate with this adapter. You will use this information during configuration of the adapter. For information, see *Configuring the Connect:Direct Server Adapter* on page 169.
If planning to use a Perimeter server with this adapter, configure the Perimeter server.
2. Create a Connect:Direct Server adapter configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Direct Server adapter. For information, see *Configuring the Connect:Direct Server Adapter* on page 169.
4. Create configurations of any Connect:Direct Server services needed to use this adapter as a PNODE. See *Related services* on page 167 in the Overview table for a list of these services and where to find more information about each.
5. Use the Connect:Direct Server services in a business process to use the adapter as a PNODE.

Configuring the Connect:Direct Server Adapter

To configure the Connect:Direct Server adapter, you must specify settings for the following fields in Gentran Integration Suite:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default)◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
Connect Direct Server Node Name	Name that will be used to identify this Connect:Direct Server adapter. Maximum 16 characters. Must match the Netmap entry for server communicating with. Required.
Connect Direct Server Port	<p>TCP/IP port number that Connect:Direct monitors for requests from remote nodes. Required.</p> <p>Note: If Gentran Integration Suite is installed on a UNIX operating system, do not select ports in the range 1-1024. These are normally reserved by the operating system.</p>
Firewall Ports	Ports used by the adapter for making outbound connections through the firewall. Optional. Valid values are hyphenated port range, a comma delimited list, or a combination of both.
Max locally initiated (pnode) sessions allowed	Maximum number of sessions the Connect:Direct Server adapter may have active at any point of time. Required. Valid value is any number up to 9999. Minimum value is 1. Default is 5.
Max remotely initiated (snode) sessions allowed	Maximum number of sessions from remote nodes that may be active at any point of time. Required. Valid value is any number up to 9999. Minimum value is 1.

Field	Description
Number of Retry attempts for establishing a session	<p>Number of times to retry creating a session. Optional. Valid value is any number up to 9999. Default is 0.</p> <p>The following errors fail without any retries:</p> <ul style="list-style-type: none"> ◆ Invalid required parameter ◆ Missing required parameter ◆ Adapter not found ◆ Remote node not in the netmap ◆ Connect:Direct unable to locate local server
Interval between session establishment attempts (minutes)	<p>Time, in minutes, to wait between attempts to create a session. Optional. Valid value is any number up to 99999999. Default is 1.</p>
Document Storage Type	<p>Defines how the document will be stored in the system. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System (Default)
Netmap Check	<p>Whether the node netmap should be checked to authorize sessions. Required. Valid values are Yes (default) and No.</p>
Buffer Size for Copy	<p>Size (in bytes) of the internal buffer used when accessing a file. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 4096 ◆ 8192 ◆ 16384 ◆ 32768 (default)
Number of Retry Attempts for Checkpoint-based copy restart	<p>Maximum number of attempts for checkpoint restart retry. Required. Valid value is any number up to 9999 (default is 3).</p> <p>Note: Checkpoint restart is only enabled for file system-based document, outbound (from Gentran Integration Suite) data transfers. (If Gentran Integration Suite is the Pnode this means business processes using the CopyTo service, if Gentran Integration Suite is the Snode this means the remote Connect:Direct Server copies documents from Gentran Integration Suite to the remote node.)</p>

Field	Description
Interval between Retry attempts (minutes)	<p>Amount of time to wait between checkpoint restart retry attempts. Required. Valid value is any number up to 99999999 (default is 1 minute).</p> <p>Note: If the Schedule_BPExpirator business process is configured to wait longer than the adapter retry interval indicates, the BPExpirator value will govern behavior. The default value for BPExpirator is 15 minutes. If the adapter retry interval is longer, the adapter value will govern behavior.</p> <p>Note: Checkpoint restart is only enabled for file system-based document, outbound (from Gentran Integration Suite) data transfers. (If Gentran Integration Suite is the Pnode this means business processes using the CopyTo service, if Gentran Integration Suite is the Snode this means the remote Connect:Direct Server copies documents from Gentran Integration Suite to the remote node.)</p>
Number of retry attempts for establishing a session	<p>Specifies the maximum number of attempts for establishing a session with a Connect:Direct node. This setting can be overridden by the value specified for Begin Session Max Retries in the Connect:Direct Server Begin Session Service. Any number up to 9999. Required.</p>
Interval between session establishment attempts (minutes)	<p>Specifies the amount of time to wait between session establishment attempts. This setting can be overridden by the value specified for Begin Session Retry Interval in the Connect Direct Server Begin Session Service. Any number up to 99999999. Required.</p>
Max Session Establishment Timeout value in Seconds	<p>Amount of time to wait before timing out an attempted connection with the Perimeter Server. Valid value is any number up to 99999999. Default is 600. Required.</p>
Max Socket Read Timeout value in Seconds:	<p>Specifies the amount of time to wait for the remote Connect:Direct Server to respond to any message sent by the Connect:Direct Server adapter. Required. Valid value is any number up to 99999999. Default is 90. When using a Perimeter server, the minimum is 90. If a value less than 90 is entered, 90 is applied.</p>
Server Start Option	<p>Whether the server will be Cold or Warm started. A Warm start maintains existing restart information. A Cold start resets the restart information. Default is Warm. Required.</p>
Connect Direct Perimeter Services Option	<p>Perimeter server and node that the Connect:Direct Server adapter will use. Optional. Valid value is any available node and perimeter server. Default is node1 & local.</p>
RunTask Business Process Name	<p>Select a business process to be run when the Connect:Direct Server adapter receives a RunJob request. Optional.</p> <p>This option is primarily for use with business processes that forward requests from one Connect:Direct node to another. You must write the business process to do the forwarding.</p> <p>Note: For information about the forwarding option, see <i>Business Process Examples</i> on page 173.</p>
RunJob Business Process Name	<p>Select a business process to be run when the Connect:Direct Server adapter receives a RunJob request. Optional.</p> <p>This option is primarily for use with business processes that forward requests from one Connect:Direct node to another. You must write the business process to do the forwarding.</p> <p>Note: For information about the forwarding option, see <i>Business Process Examples</i> on page 173.</p>

Field	Description
Max Run Task Forwarding Timeout value in Seconds:	The amount of time (in seconds) to wait before timing out a session that forwards a RunTask request. Required. Default is 300. Valid value is any natural number.
Max Run Job Forwarding Timeout value in Seconds:	The amount of time (in seconds) to wait before timing out a session that forwards a RunJob request. Required. Default is 30. Valid value is any natural number.
Extractable Count	The number of times the message can be extracted. Optional. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer.
Extractable For	The length of time (in days, hours and minutes) the message can be extracted. Optional. Cannot be specified in conjunction with Extractable or Extractable Count. Format is 'dddhhmm'.
Extractable	A yes or no value indicating if this message can be extracted. Optional. Cannot be specified in conjunction with Extractable Count or Extractable For. Valid values are Yes and No.
Encryption using Secure+	Whether Secure+ is enabled. Required. Valid values are Enabled and Disabled. Default is Disabled. Note: The following four fields are displayed only if Enabled is selected.
CA Certificates	Drop-down menu that contains a list of trusted Certificate Authority public certificates. Only displayed if "Encryption using Secure+" is enabled. Required. if "Encryption using Secure+" is enabled. Valid values are all CA certificates in this Gentran Integration Suite installation.
System Certificate	List of private keys and public certificates signed by the trading partner trusted certificate authority. This represents the certificate for the local node (this adapter). Only displayed if Encryption using Secure+ is enabled. Required if Encryption using Secure+ is enabled. Valid values are all system certificates in this Gentran Integration Suite installation.
Cipher Suites	Indicates cipher suites available to support varying encryption levels. Multiple selections allowed. Selections can be rank ordered. Only displayed if "Encryption using Secure+" is enabled. Required if "Encryption using Secure+" is enabled. Valid values are all available cipher suites in this Gentran Integration Suite installation.
SSL or TLS	Indicates if Secure+ is enabled using SSL or TLS. Only displayed if "Encryption using Secure+" is enabled. Required if "Encryption using Secure+" is enabled. Valid values are SSL and TLS. Default is SSL.
Enable Netmap Node Override	Specifies whether node netmap security settings should override the adapter security settings. Required. Valid values are Yes and No. Default is No.
Connect:Direct Server Nodes	Define information for each remote Connect:Direct node that adapter will communicate with. Equivalent to creating Netmap entries on a Connect:Direct Server. Optional.

Reserved Words

The following reserved words are used with the Connect:Direct Server adapter for COPY and SUBMIT statements in a Connect:Direct process script:

`businessprocess`

mailbox

These must be used at the beginning of the path for the target (or source) of the copy/submit in Gentran Integration Suite. For example:

```
/businessprocess/myspecialBPname/targetdocName
```

or

```
/mailbox/customers/CustomerXYZ/inbound/purchaseOrder.txt
```

could be used in the TO or FROM clause of a COPY statement in a Connect:Direct process script.

Additionally, there is a mailbox called /DeadLetter that is reserved for use by Gentran Integration Suite.

Business Process Examples

The following samples illustrate how information from a Connect:Direct Server adapter might be used in a business process. The adapter itself is not used in business processes, but information provided during adapter configuration is used by its related Connect:Direct Server services.

The following sample business process illustrates using the Connect:Direct Server CopyTo service in a business process. The business process copies the current primary document to the remote Connect:Direct Server, which is identified with a session token. The (optional) Sysopts in this example are in a format valid for Connect:Direct Windows only.

```
<process name=" CopyToExample">
  <sequence name="CDServerCopyToGIS">
    <!-- Begin Server Session -->
      <operation name="Connect:Direct Server Begin Session Service">
        <participant name="CDServerBeginSession"/>
        <output message="BeginSession">
          <assign to="." from="*"></assign>
          <assign to="LocalCDNodeName">TEST1_CDSEVER</assign>
          <assign to="RemoteCDNodeName">REMOTECDNODE1</assign>
          <assign to="RemoteUserId">user1</assign>
          <assign to="RemotePasswd">password1</assign>
        </output>
        <input message="inmsg">
          <assign to="BeginSessionResults" from="*"></assign>
        </input>
      </operation>
      <operation name="Connect:Direct Server Copy To Service">
        <participant name="CDServerCopyTo"/>
        <output message="CopyTo">
          <assign to="." from="*"></assign>
          <assign to="SessionToken"
            from="//BeginSessionResults/SessionToken/node()"></assign>
          <assign to="RemoteFileName">sample.txt</assign>
          <assign to="BinaryMode">no</assign>
          <assign to="Sysopts">datatype(text) xlate(no)</assign>
        </output>
        <input message="inmsg">
          <assign to="CopyToResults" from="*"></assign>
        </input>
      </operation>
    </sequence>
  </process>
```

```

<!-- End Server Session -->
  <operation name="Connect:Direct Server End Session Service">
    <participant name="CDServerEndSession"/>
    <output message="EndSession">
      <assign to="." from="*"></assign>
      <assign to="SessionToken"
        from="//BeginSessionResults/SessionToken/node()"></assign>
    </output>
    <input message="inmsg">
      <assign to="EndSessionResults" from="*"></assign>
    </input>
  </operation>
</sequence>
</process>

```

Forwarding a Request from One Connect:Direct Node to Another

The following sample business process uses the Connect:Direct Server Run Job service to forward a request:

```

<process name = "CDInterop_testRunJobForward">
  <sequence name = "CDBeginSession">
    <operation name="Connect:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">aaa.bbb</assign>
        <assign to="RemoteCDNodeName">ZZ.AAA.ZZZZZ</assign>
        <assign to="RemoteUserId">username</assign>
        <assign to="RemotePasswd">password</assign>
      </output>
      <input message="inmsg">
        <assign to="BeginSessionResults" from="*"></assign>
      </input>
    </operation>
    <operation name="Connect:Direct Server Run Job Service">
      <participant name="CDServerRunJob"/>
      <output message="RunJob">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
          from="//BeginSessionResults/SessionToken/node()"></assign>
        <assign to="Sysopts">AAAA1.jcl</assign>
      </output>
      <input message="inmsg">
        <assign to="Run_Job_Results" from="*"></assign>
      </input>
    </operation>
  <operation name="CONNECT:Direct Server BP Response Service">
    <participant name="CDServerTestBPResponse1"/>
    <output message="BPResponseParams">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="BPResponseResults" from="*"></assign>
    </input>
  </operation>
</sequence>
</process>

```

```

</operation>

<operation name="Connect:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="OnFaultHandler" from="*"></assign>
  </input>
</operation>
<!-- Cleanup when bad things happen (exceptions) -->
<onFault>
  <sequence name="OnFault">
<operation name="CONNECT:Direct Server BP Response Service">
  <participant name="CDServerTestBPResponse1"/>
  <output message="BPResponseParams">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="BPResponseResults" from="*"></assign>
  </input>
</operation>
</sequence>
</onFault>
</sequence>
</process>

<!-- End the Session which was begun before the fault code -->
<operation name="Connect:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="EndSessionResults" from="*"></assign>
  </input>
</operation>
</sequence>
</onFault>
</sequence>
</process>

```

The following sample business process uses the Connect:Direct Server Run Task service to forward a request:

```

<process name = "CDInterop_testRunTaskForward">
  <sequence name = "CDBeginSession">
    <operation name="Connect:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">aaaa.bbbb</assign>
        <assign to="RemoteCDNodeName">ZZ.AAA.ZZZZZ</assign>
        <assign to="RemoteUserId">username</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

    <assign to="RemotePasswd">password</assign>
  </output>
  <input message="inmsg">
    <assign to="BeginSessionResults" from="*"></assign>
  </input>
</operation>
<operation name="Connect:Direct Server Run Task Service">
  <participant name="CDServerRunTask"/>
  <output message="RunTask">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
    <assign to="Sysopts">AAAAA</assign>
  </output>
  <input message="inmsg">
    <assign to="Run_TASK_Results" from="*"></assign>
  </input>
</operation>
<operation name="CONNECT:Direct Server BP Response Service">
  <participant name="CDServerTestBPResponse1"/>
  <output message="BPResponseParams">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="BPResponseResults" from="*"></assign>
  </input>
</operation>

<operation name="Connect:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="OnFaultHandler" from="*"></assign>
  </input>
</operation>
<!-- Cleanup when bad things happen (exceptions) -->
<onFault>
  <sequence name="OnFault">
<operation name="CONNECT:Direct Server BP Response Service">
  <participant name="CDServerTestBPResponse1"/>
  <output message="BPResponseParams">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="BPResponseResults" from="*"></assign>
  </input>
</operation>

<!-- End the Session which was begun before the fault code -->
<operation name="Connect:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">

```



```
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
           from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="EndSessionResults" from="*"></assign>
  </input>
</operation>
</sequence>
</onFault>
</sequence>
</process>
```

Activity Types for This Service

This adapter reports the following activities to the Gentran Integration Suite Services Controller for service and adapter monitoring:

- Session — lists any current session that the adapter has and details activities of the session
- Copy — shows movement of a file between the Connect:Direct Server adapter and a remote Connect:Direct node

Connect:Direct Server Adapter (Build 4309 or higher)

The following table provides an overview of the Connect:Direct Server adapter:

System Name	Connect:Direct Server Adapter
Graphical Process Modeler (GPM) category	Not configurable in the GPM.
Description	This adapter receives and processes requests from remote Connect:Direct nodes.
Business usage	Use this adapter to: <ul style="list-style-type: none">◆ Mediate requests originating from a business process and going to an external trading partner using a (non-application) Connect:Direct node or another Connect:Direct Server adapter.◆ Accept requests from a Connect:Direct node (or another Connect:Direct Server adapter) to copy to or from a mailbox or business process.◆ Forward requests from one remote Connect:Direct node to another
Usage example	A trading partner submits a Connect:Direct process on their local system. The process that runs must initiate a session with the application Connect:Direct node and copy a file. The Connect:Direct Server adapter allows the session and responds to the copy request.
Preconfigured?	No preconfigured copy of the adapter is provided.
Requires third party files?	The following files external to your application are required: <ul style="list-style-type: none">◆ Connect:Direct Server◆ Certicom SSL Library (included with your application)◆ JZLIB (included with your application)◆ CDJava.jar (build 81) (included with your application)
Platform availability	All supported platforms for your application

Related services	<p>To use the Connect:Direct Server adapter as a PNODE, you must configure and use one or more of the following services:</p> <ul style="list-style-type: none"> ◆ Connect:Direct Server Begin Session Service ◆ Connect:Direct Server BP Response Service ◆ Connect:Direct Server CopyFrom Service ◆ Connect:Direct Server CopyTo Service ◆ Connect:Direct Server End Session Service ◆ Connect:Direct Server Run Job Service ◆ Connect:Direct Server Run Task Service ◆ Connect:Direct Server Submit Service <p>The above services can be used as part of a business process. When invoked, they use a Connect:Direct Server adapter to perform specified activities with a remote Connect:Direct node. The adapter and remote node context are provided by a session token created by the Connect:Direct Server Begin Session Service. Subsequent services in the business process refer to this session token.</p>
Application requirements	<p>The following are optional choices for interoperability with Connect:Direct.</p> <ul style="list-style-type: none"> ◆ Connect:Direct Windows 4.1.00 (with patch 26) or later ◆ Connect:Direct UNIX 3.6.00 or later ◆ Connect:Direct OS/390 4.4.00 or later ◆ Connect:Direct HP Non-Stop ◆ Connect:Direct Select
Initiates business processes?	Initiates the business process specified by the remote Connect:Direct node or a business process selected as a target for a copy.
Invocation	The adapter is started when your application is started.
Business process context considerations	None
Returned status values	None

Restrictions	<p>The Connect:Direct Server adapter includes parameters to govern checkpoint/restart behavior if either the application or the remote node fails. If the remote Connect:Direct Server or the network fails during a copy operation, the adapter goes into a retry mode using the values specified at configuration. The adapter waits the configured amount of time and then resumes the copy.</p> <p>The length of time the adapter waits to resume the copy is impacted by two values:</p> <ul style="list-style-type: none"> ◆ Interval between Retry attempts (seconds), as specified in the adapter configuration. Default is 1 minute. ◆ Frequency defined by the scheduled system business process BPEXpirator. BPEXpirator is used in the Wait service. The Wait service is used when a system failure interrupts a Connect:Direct copy operation. Default is 15 minutes. <p>Whichever value is longer governs behavior. By coordinating the adapter retry interval and the BPEXpirator schedule, you can configure a window of time appropriate for your needs. Your application enables automatic restart when the recovery level is set to Auto-Resume. Checkpoint restart is only enabled for file system-based document, outbound (from your application) data transfers. (If your application is the Pnode, this means business processes using the CopyTo Service. If your application is the Snode, this means the remote Connect:Direct Server copies documents from your application to the remote node). All other restarted copy operations are started from the beginning of the document.</p>
Persistence level	None
Testing considerations	Debug information for this adapter can be found in the Connect:Direct Server adapter log files.

Implementing the Connect:Direct Server Adapter

To implement the Connect:Direct Server adapter, complete the following tasks:

1. Collect information about any remote Connect:Direct Servers that will communicate with this adapter. You will use this information during configuration of the adapter. For information, see *Configuring the Connect:Direct Server Adapter*.
If planning to use a Perimeter server with this adapter, configure the Perimeter server.
2. Create a Connect:Direct Server adapter configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Direct Server adapter. For information, see *Configuring the Connect:Direct Server Adapter*.
4. Create configurations of any Connect:Direct Server services needed to use this adapter as a PNODE. See *Related services* in the Overview table for a list of these services and where to find more information about each.
5. Use the Connect:Direct Server services in a business process to use the adapter as a PNODE.

Configuring the Connect:Direct Server Adapter

To configure the Connect:Direct Server adapter, you must specify settings for the following fields in your application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default)◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
Connect Direct Server Node Name	Name that will be used to identify this Connect:Direct Server adapter. Maximum 16 characters. Must match the Netmap entry for server communicating with. Required.
Connect Direct Server Port	<p>TCP/IP port number that Connect:Direct monitors for requests from remote nodes. Required.</p> <p>Note: If your application is installed on a UNIX operating system, do not select ports in the range 1-1024. These are normally reserved by the operating system.</p>
Firewall Ports	Ports used by the adapter for making outbound connections through the firewall. Optional. Valid values are hyphenated port range, a comma delimited list, or a combination of both.
Max locally initiated (pnode) sessions allowed	<p>Maximum number of sessions the Connect:Direct Server adapter may have active at any point of time. Required.</p> <p>May be overridden at the node level. If a node-level value is configured, the effective session limit is the smaller of the two values: the limit for the adapter and the session limit for the remote node.</p> <p>Valid value: Any number through 9999. Minimum is 1. Default is 5.</p>
Max remotely initiated (snode) sessions allowed	<p>Maximum number of sessions from remote nodes that may be active at any point of time. Required.</p> <p>May be overridden at the node level. If a node-level value is configured, the effective session limit is the smaller of the two values: the limit for the adapter and the session limit for the remote node.</p> <p>Valid value: Any number through 9999. Minimum is 1. Default is 5.</p>

Field	Description
Number of Retry attempts for establishing a session	<p>Number of times to retry creating a session. Optional. Valid value is any number up to 9999. Default is 0.</p> <p>The following errors fail without any retries:</p> <ul style="list-style-type: none"> ◆ Invalid required parameter ◆ Missing required parameter ◆ Adapter not found ◆ Remote node not in the netmap ◆ Connect:Direct unable to locate local server
Interval between session establishment attempts (minutes)	<p>Time, in minutes, to wait between attempts to create a session. Optional. Valid value is any number up to 99999999. Default is 1.</p>
Document Storage Type	<p>Defines how the document will be stored in the system. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System (Default)
Netmap Check	<p>Whether the node netmap should be checked to authorize sessions. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ Check both node name and IP address – Searches the CDSA network map for an entry corresponding to the node name of the remote client. If one exists, the IP address is obtained and compared to the IP address of the remote node making the request. ◆ Check node name only – Searches the CDSA network map for an entry corresponding to the node name of the remote client. ◆ No (Default) – Disables netmap validation <p>Note: Connect:Direct Server adapters that were configured before availability of the “Check node name only” option are not affected. The previous “Yes” value corresponds to “Check both node name and IP address.”</p>
Buffer Size for Copy	<p>Size (in bytes) of the internal buffer used when accessing a file. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 4096 ◆ 8192 ◆ 16384 ◆ 32768 (default)
Number of Retry Attempts for Checkpoint-based copy restart	<p>Maximum number of attempts for checkpoint restart retry. Required. Valid value is any number up to 9999 (default is 3).</p> <p>Note: Checkpoint restart is only enabled for file system-based document, outbound (from your application) data transfers. (If your application is the Pnode this means business processes using the CopyTo service. If your application is the Snode, this means the remote Connect:Direct Server copies documents from your application to the remote node.)</p>

Field	Description
Interval between Retry attempts (minutes)	<p>Amount of time to wait between checkpoint restart retry attempts. Required. Valid value is any number up to 99999999 (default is 1 minute).</p> <p>Note: If the Schedule_BPExpirator business process is configured to wait longer than the adapter retry interval indicates, the BPExpirator value will govern behavior. The default value for BPExpirator is 15 minutes. If the adapter retry interval is longer, the adapter value will govern behavior.</p> <p>Note: Checkpoint restart is only enabled for file system-based document, outbound (from your application) data transfers. (If your application is the Pnode, this means business processes using the CopyTo service. If your application is the Snode, this means the remote Connect:Direct Server copies documents from your application to the remote node.)</p>
Number of short-term session retry attempts	<p>Specifies the maximum number of short-term session retry attempts that should be made when failures to connect occur.</p> <p>This setting can be overridden by the value specified for ShortTermMaxRetries in the Connect:Direct Server Begin Session service.</p> <p>Valid value: Any number up through 9999. Default is 0. Required.</p> <p>Short-term retry activity is logged (if logging enabled).</p>
Interval between short-term session attempts (seconds)	<p>Specifies the amount of time to wait between short-term session establishment attempts. This setting can be overridden by the value specified for ShortTermRetryInterval in the Connect Direct Server Begin Session service.</p> <p>Valid value: Any number through 99999999. Default is 5. Required.</p> <p>Short-term retry activity is logged (if logging enabled).</p>
Number of retry attempts for establishing a session	<p>Specifies the maximum number of attempts for establishing a session with a Connect:Direct node. This setting can be overridden by the value specified for Begin Session Max Retries in the Connect:Direct Server Begin Session Service. Any number up to 9999. Required.</p>
Interval between session establishment attempts (minutes)	<p>Specifies the amount of time to wait between session establishment attempts. This setting can be overridden by the value specified for Begin Session Retry Interval in the Connect Direct Server Begin Session Service. Any number up to 99999999. Required.</p>
Max Session Establishment Timeout value in Seconds	<p>Amount of time to wait before timing out an attempted connection with the Perimeter Server. Valid value is any number up to 99999999. Default is 600. Required.</p>
Max Socket Read Timeout value in Seconds:	<p>Specifies the amount of time to wait for the remote Connect:Direct Server to respond to any message sent by the Connect:Direct Server adapter. Required. Valid value is any number up to 99999999. Default is 90. When using a Perimeter server, the minimum is 90. If a value less than 90 is entered, 90 is applied.</p>
Server Start Option	<p>Whether the server will be Cold or Warm started. A Warm start maintains existing restart information. A Cold start resets the restart information. Default is Warm. Required.</p>
Connect Direct Perimeter Services Option	<p>Perimeter server and node that the Connect:Direct Server adapter will use. Optional. Valid value is any available node and perimeter server. Default is node1 & local.</p>

Field	Description
RunTask Business Process Name	<p>Select a business process to be run when the Connect:Direct Server adapter receives a RunJob request. Optional.</p> <p>This option is primarily for use with business processes that forward requests from one Connect:Direct node to another. You must write the business process to do the forwarding.</p> <p>Note: For information about the forwarding option, see <i>Business Process Examples</i> on page 8.</p>
RunJob Business Process Name	<p>Select a business process to be run when the Connect:Direct Server adapter receives a RunJob request. Optional.</p> <p>This option is primarily for use with business processes that forward requests from one Connect:Direct node to another. You must write the business process to do the forwarding.</p> <p>Note: For information about the forwarding option, see <i>Business Process Examples</i> on page 8.</p>
Max Run Task Forwarding Timeout value in Seconds:	The amount of time (in seconds) to wait before timing out a session that forwards a RunTask request. Required. Default is 300. Valid value is any natural number.
Max Run Job Forwarding Timeout value in Seconds:	The amount of time (in seconds) to wait before timing out a session that forwards a RunJob request. Required. Default is 30. Valid value is any natural number.
Extractable Count	The number of times the message can be extracted. Optional. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer.
Extractable For	The length of time (in days, hours and minutes) the message can be extracted. Optional. Cannot be specified in conjunction with Extractable or Extractable Count. Format is 'dddhhmm'.
Extractable	A yes or no value indicating if this message can be extracted. Optional. Cannot be specified in conjunction with Extractable Count or Extractable For. Valid values are Yes and No.
Encryption using Secure+	<p>Whether Secure+ is enabled. Required. Valid values are Enabled and Disabled. Default is Disabled.</p> <p>Note: The following four fields are displayed only if Enabled is selected.</p>
CA Certificates	Drop-down menu that contains a list of trusted Certificate Authority public certificates. Only displayed if "Encryption using Secure+" is enabled. Required. if "Encryption using Secure+" is enabled. Valid values are all CA certificates in this installation of your application.
System Certificate	List of private keys and public certificates signed by the trading partner trusted certificate authority. This represents the certificate for the local node (this adapter). Only displayed if Encryption using Secure+ is enabled. Required if Encryption using Secure+ is enabled. Valid values are all system certificates in this installation of your application.
Certificate Common Name	<p>Value to be compared with the Common Name (CN) field from the remote node's handshake during TLS/SSL handshake in order to enhance certificate authentication. Only displayed if "Encryption using Secure+" is enabled. May be overridden at the node level.</p> <p>Note: To see the actual CNs being compared during the handshake, set the Perimeter Services logging level to ALL.</p>

Field	Description
Cipher Suites	Indicates cipher suites available to support varying encryption levels. Multiple selections allowed. Selections can be rank ordered. Only displayed if "Encryption using Secure+" is enabled. Required if "Encryption using Secure+" is enabled. Valid values are all available cipher suites in this installation of your application.
SSL or TLS	Indicates if Secure+ is enabled using SSL or TLS. Only displayed if "Encryption using Secure+" is enabled. Required if "Encryption using Secure+" is enabled. Valid values are SSL and TLS. Default is SSL.
Enable Netmap Node Override	Specifies whether node netmap security settings should override the adapter security settings. Required. Valid values are Yes and No. Default is No.
Connect:Direct Server Nodes	Define information for each remote Connect:Direct node that adapter will communicate with. Equivalent to creating Netmap entries on a Connect:Direct Server. Optional.

Reserved Words

The following reserved words are used with the Connect:Direct Server adapter for COPY and SUBMIT statements in a Connect:Direct process script:

```
businessprocess
mailbox
```

These must be used at the beginning of the path for the target (or source) of the copy/submit in your application. For example:

```
/businessprocess/myspecialBPname/targetdocName
```

or

```
/mailbox/customers/CustomerXYZ/inbound/purchaseOrder.txt
```

could be used in the TO or FROM clause of a COPY statement in a Connect:Direct process script.

Additionally, there is a mailbox called `/DeadLetter` that is reserved for use by your application.

Business Process Examples

The following samples illustrate how information from a Connect:Direct Server adapter might be used in a business process. The adapter itself is not used in business processes, but information provided during adapter configuration is used by its related Connect:Direct Server services.

The following sample business process illustrates using the Connect:Direct Server CopyTo service in a business process. The business process copies the current primary document to the remote Connect:Direct Server, which is identified with a session token. The (optional) Sysopts in this example are in a format valid for Connect:Direct Windows only.

```
<process name=" CopyToExample">
  <sequence name="CDServerCopyToGIS">
    <!-- Begin Server Session -->
      <operation name="Connect:Direct Server Begin Session Service">
```

```

    <participant name="CDServerBeginSession"/>
    <output message="BeginSession">
      <assign to="." from="*"></assign>
      <assign to="LocalCDNodeName">TEST1_CDSEVER</assign>
      <assign to="RemoteCDNodeName">REMOTECDNODE1</assign>
      <assign to="RemoteUserId">user1</assign>
      <assign to="RemotePasswd">password1</assign>
    </output>
    <input message="inmsg">
      <assign to="BeginSessionResults" from="*"></assign>
    </input>
  </operation>
  <operation name="Connect:Direct Server Copy To Service">
    <participant name="CDServerCopyTo"/>
    <output message="CopyTo">
      <assign to="." from="*"></assign>
      <assign to="SessionToken"
        from="//BeginSessionResults/SessionToken/node()"></assign>
      <assign to="RemoteFileName">sample.txt</assign>
      <assign to="BinaryMode">no</assign>
      <assign to="Sysopts">datatype(text) xlate(no)</assign>
    </output>
    <input message="inmsg">
      <assign to="CopyToResults" from="*"></assign>
    </input>
  </operation>
<!-- End Server Session -->
  <operation name="Connect:Direct Server End Session Service">
    <participant name="CDServerEndSession"/>
    <output message="EndSession">
      <assign to="." from="*"></assign>
      <assign to="SessionToken"
        from="//BeginSessionResults/SessionToken/node()"></assign>
    </output>
    <input message="inmsg">
      <assign to="EndSessionResults" from="*"></assign>
    </input>
  </operation>
</sequence>
</process>

```

Forwarding a Request from One Connect:Direct Node to Another

The following sample business process uses the Connect:Direct Server Run Job service to forward a request:

```

<process name = "CDInterop_testRunJobForward">
  <sequence name = "CDBeginSession">
    <operation name="Connect:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">aaa.bbb</assign>
        <assign to="RemoteCDNodeName">ZZ.AAA.ZZZZZ</assign>
        <assign to="RemoteUserId">username</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

    <assign to="RemotePasswd">password</assign>
  </output>
  <input message="inmsg">
    <assign to="BeginSessionResults" from="*"></assign>
  </input>
</operation>
<operation name="Connect:Direct Server Run Job Service">
  <participant name="CDServerRunJob"/>
  <output message="RunJob">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
    <assign to="Sysopts">AAAA1.jcl</assign>
  </output>
  <input message="inmsg">
    <assign to="Run_Job_Results" from="*"></assign>
  </input>
</operation>
<operation name="CONNECT:Direct Server BP Response Service">
  <participant name="CDServerTestBPResponse1"/>
  <output message="BPResponseParams">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="BPResponseResults" from="*"></assign>
  </input>
</operation>

<operation name="Connect:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="OnFaultHandler" from="*"></assign>
  </input>
</operation>
<!-- Cleanup when bad things happen (exceptions) -->
<onFault>
  <sequence name="OnFault">
<operation name="CONNECT:Direct Server BP Response Service">
  <participant name="CDServerTestBPResponse1"/>
  <output message="BPResponseParams">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="BPResponseResults" from="*"></assign>
  </input>
</operation>

<!-- End the Session which was begun before the fault code -->
<operation name="Connect:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">

```

```

    <assign to="." from="*"></assign>
    <assign to="SessionToken"
           from="//BeginSessionResults/SessionToken/node()"></assign>
</output>
<input message="inmsg">
    <assign to="EndSessionResults" from="*"></assign>
</input>
</operation>
</sequence>
</onFault>
</sequence>
</process>

```

The following sample business process uses the Connect:Direct Server Run Task service to forward a request:

```

<process name = "CDInterop_testRunTaskForward">
  <sequence name = "CDBeginSession">
    <operation name="Connect:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">aaaa.bbbb</assign>
        <assign to="RemoteCDNodeName">ZZ.AAA.ZZZZZ</assign>
        <assign to="RemoteUserId">username</assign>
        <assign to="RemotePasswd">password</assign>
      </output>
      <input message="inmsg">
        <assign to="BeginSessionResults" from="*"></assign>
      </input>
    </operation>
    <operation name="Connect:Direct Server Run Task Service">
      <participant name="CDServerRunTask"/>
      <output message="RunTask">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
               from="//BeginSessionResults/SessionToken/node()"></assign>
        <assign to="Sysopts">AAAAA</assign>
      </output>
      <input message="inmsg">
        <assign to="Run_TASK_Results" from="*"></assign>
      </input>
    </operation>
    <operation name="CONNECT:Direct Server BP Response Service">
      <participant name="CDServerTestBPResponse1"/>
      <output message="BPResponseParams">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="BPResponseResults" from="*"></assign>
      </input>
    </operation>

    <operation name="Connect:Direct Server End Session Service">
      <participant name="CDServerEndSession"/>

```

```

    <output message="EndSession">
      <assign to="." from="*"></assign>
      <assign to="SessionToken"
        from="//BeginSessionResults/SessionToken/node()"></assign>
    </output>
    <input message="inmsg">
      <assign to="OnFaultHandler" from="*"></assign>
    </input>
  </operation>
<!-- Cleanup when bad things happen (exceptions) -->
  <onFault>
    <sequence name="OnFault">
<operation name="CONNECT:Direct Server BP Response Service">
  <participant name="CDServerTestBPResponse1"/>
  <output message="BPResponseParams">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="BPResponseResults" from="*"></assign>
  </input>
</operation>

  <!-- End the Session which was begun before the fault code -->
  <operation name="Connect:Direct Server End Session Service">
    <participant name="CDServerEndSession"/>
    <output message="EndSession">
      <assign to="." from="*"></assign>
      <assign to="SessionToken"
        from="//BeginSessionResults/SessionToken/node()"></assign>
    </output>
    <input message="inmsg">
      <assign to="EndSessionResults" from="*"></assign>
    </input>
  </operation>
</sequence>
</onFault>
</sequence>
</process>

```

Activity Types for This Service

This adapter reports the following activities to the Services Controller of your application for service and adapter monitoring:

- Session — lists any current session that the adapter has and details activities of the session
- Copy — shows movement of a file between the Connect:Direct Server adapter and a remote Connect:Direct node

Connect:Direct Server Begin Session Service

The following table provides an overview of the Connect:Direct Server Begin Session service:

System Name	Begin Session Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Direct
Description	Used to begin a session with a remote Connect:Direct node.
Business usage	Use this service to initiate a session on a remote Connect:Direct node. The service then uses the Connect:Direct Server adapter to send outbound requests to the remote node. For information about the adapter, see <i>Connect:Direct Server Adapter</i> .
Usage example	A Application business process includes a file as primary document that must be copied to a remote Connect:Direct node. Before the copy operation can begin, a session must be established with the remote Connect:Direct node by including this service in the business process.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms
Related services	<ul style="list-style-type: none">◆ Connect:Direct Server End Session Service (must be used to end the session started by this service)◆ Connect:Direct Server Adapter
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No
Invocation	This service can run by a business process to initiate an activity with a remote Connect:Direct node. The business process running this service must be started by a user with permission to initiate a session on the remote Connect:Direct node.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified below. ◆ Invalid UserId Error – User specified is not valid. ◆ Invalid Password Error – Password is not valid. ◆ Net Map Check Error – The Net Map checking failed and you should check the Net Map entries. ◆ Connection TimeOut Error – Could not establish connection within 60 seconds. ◆ Connect:Direct Server Error – A generic error associated with the Connect:Direct server has occurred. ◆ Connect:Direct Error – A generic error associated with the Connect:Direct has occurred. ◆ Connect:Direct Service Error – A generic error associated with the Connect:Direct service has occurred.
Restrictions	None
Persistence level	None
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Server Begin Session Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server Begin Session service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Create a Connect:Direct Server Begin Session service configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Direct Server Begin Session service. For information, see *Configuring the Connect:Direct Server Begin Session Service* on page 191.
4. Use the Connect:Direct Server Begin Session service in a business process.

Configuring the Connect:Direct Server Begin Session Service

To configure the Connect:Direct Server Begin Session service, you must specify settings for the following fields in Application.

Note: Field names in the GPM are provided in parentheses when different from the Administration menu.

Field	Description
Name	Name of the service configuration. Required.

Field	Description
Description	Description of service. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Server Begin Session service:

Field	Description
Number of Retry attempts for establishing a session (BeginSessionMaxRetries)	<p>Number of times to retry creating a session. Optional. When specified here, the value overrides the values set in the Connect:Direct Server adapter configuration. Valid value is any number up to 9999. Default is 0.</p> <p>The following errors fail without any retries:</p> <ul style="list-style-type: none"> ◆ Invalid required parameter ◆ Missing required parameter ◆ Adapter not found ◆ Remote node not in the netmap ◆ Connect:Direct unable to locate local server
Interval between session establishment attempts (minutes) (BeginSessionRetryInterval)	<p>Time, in minutes, to wait between attempts to create a session. Optional. When specified here, the value overrides the values set in the Connect:Direct Server adapter configuration. Valid value is any number up to 99999999. Default is 1.</p>
LocalCDNodeName	<p>Specifies the name of the local Connect: Direct Server adapter. List includes all configurations of the Connect:Direct Server adapter. Required.</p>
PlexClass	<p>Specifies the PlexClass to be used on the remote Connect:Direct server node. Optional.</p>
ProcessName	<p>Specifies a user-designated name for the Application business process that initiated the Connect:Direct Server Begin Session service. Optional. If not specified, the process name is generated dynamically. Maximum eight characters.</p> <p>Note: If a ProcessName is specified, hyperlinks from the Connect:Direct Browser interface to Application are not active.</p>

Field	Description
RemoteCDNodeName	Specifies the name of a Remote Connect:Direct Server node. This name must be same as the name of one of the nodes in the netmap of the adapter identified by LocalCDNodeName. Required.
RemotePasswd	Specifies the password to be used on Remote Connect:Direct Server Node. Optional.
RemoteUserId	Specifies the UserId to be used on remote Connect:Direct server node. Required except when: <ol style="list-style-type: none"> 1. Remote password not provided in the invoked BP. 2. Remote Connect:Direct Server is configured to accept an inbound connection from a designated PNodeName (Specified Connect:Direct server adapter name in the invoked BP) and PNodeId (the user executing the BP)

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Server Begin Session service to the business process:

Parameter	Description
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval
SessionBeginTime	Specifies the date and time when the session was established.

Business Process Example

The following business process example uses the Obscure Data -Process Data Values service to obscure the password in BPML. This example business process will only succeed if an instance of the Obscure Data -Process Data Values service called "CDClientObscureParameter" has already been created with "cinterop" and "*****" (the password value) as values:

```

<process name="CDInterop_SessionBegin">

  <sequence name="CDServerSessionBeginGIS">
<operation name="Obscure Password">
  <participant name="CDClientObscureParameter"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

  <!-- Begin Server Session -->
  <operation name="Connect:Direct Server Begin Session Service">
    <participant name="CDServerBeginSession"/>
    <output message="BeginSession">
      <assign to="." from="*"></assign>
      <assign to="LocalCDNodeName">TEST1_CDSEVER</assign>
      <assign to="RemoteCDNodeName">ZZ.AAA.ZZZZZ</assign>
      <assign to="RemoteUserId">auser</assign>
      <assign to="RemotePasswd" from="revealObscured(cinterop)"></assign>
    </output>
    <input message="inmsg">
      <assign to="BeginSessionResults" from="*"></assign>
    </input>
  </operation>

  <!-- End Server Session -->
  <operation name="CONNECT:Direct Server End Session Service">
    <participant name="CDServerEndSession"/>
    <output message="EndSession">
      <assign to="." from="*"></assign>
      <assign to="SessionToken"
        from="//BeginSessionResults/SessionToken/node()"></assign>
    </output>
    <input message="inmsg">
      <assign to="EndSessionResults" from="*"></assign>
    </input>
  </operation>

  </sequence>
</process>

```

Connect:Direct Server BP Response Service

The following table provides an overview of the Connect:Direct Server BP Response service:

System Name	Connect:Direct Server BP Response Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Direct
Description	This service is used to call back from a business process to a Connect:Direct Server adapter.
Business usage	<p>This service is used as the last operation in a business process launched from a Connect:Direct Server adapter in response to a request from a remote Connect:Direct Server to copy-from-BP. The service performs a call-back on the Connect:Direct Server adapter that launched the business process to convey the primary document which is used as the response to send to the remote Connect:Direct server. (This is similar to the HTTP Respond service, which performs a callback on the HTTP Server adapter).</p> <p>This service uses session information that is pre-placed in the process data by the Connect:Direct Server adapter that launches the business process. Therefore the service can only be used in business processes launched by a Connect:Direct Server adapter.</p>
Usage example	A remote Connect:Direct server initiates a session with a Connect:Direct Server adapter in Application and requests to copy (retrieve) the final primary document resulting from a business process launched by the Connect:Direct Server adapter. The name of the business process is provided in the request from the remote Connect:Direct node. The business process must be created in Application and have the BP Response Service as the last operation, so that the service can call back the adapter to respond to the request from the remote Connect:Direct node.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms
Related services	Connect:Direct Server adapter. The Connect:Direct Server BP Response service can only be used in business processes launched by the Connect:Direct Server adapter.
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No

Invocation	Important: This service can only be used in a business process launched by a Connect:Direct Server adapter, in response to a copy-from-business-process request from a remote Connect:Direct server. This service must be the last operation in such a business process. This service does not have any parameters to be configured.
Business process context considerations	None
Returned status values	None
Restrictions	Can only be used in a business process launched by a Connect:Direct Server adapter in response to a “copy-from BP” request from a remote Connect:Direct server. This service must be the last operation in the business process. This service does not have any configurable parameters.
Persistence level	Default
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Server BP Response Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server BP Response service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Create a Connect:Direct Server BP Response service configuration. For information, see *Managing Services and Adapters*.
3. Use the Connect:Direct Server BP Response service in a business process.

Configuring the Connect:Direct Server BP Response Service

There is no configuration required for the Connect:Direct Server BP Response service.

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Server BP Response service:

Parameter	Description
Callback_SessionToken	Specifies the identifier for the session established between two Connect:Direct server nodes. Required.

Business Process Example

The following example illustrates how the Connect:Direct Server BP Response service is used in a business process:

```
<process name="CDInterop_SnodeCallback_BP">
  <!-- A simple BP that may be kicked off at Snode during CDServer Copy
  from BP test (Application as Snode).
  !!NOTE!! :
  This BP may ONLY be launched by the BusinessProcessInitiator class;
  will fail if launched using execution manager.
  -->
  <sequence name="CDServerSnodeCallbackBP">

    <!-- Read a file from disk using the file-system adapter which
    will become the primary document-->

    <operation name="File System Adapter">
      <participant name="CDInteropTestFSA"/>
      <output message="FileSystemParams">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="bootstrap">>false</assign>
        <assign to="collectionFolder">/install_dir/foo/bar/in</assign>
        <assign to="useSubFolders">>false</assign>
        <assign to="filter" from="Samplefile.txt"/> <!-- file to read -->
        <assign to="deleteAfterCollect">>false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="FileSystemResults">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <!-- Perform callback on the adapter using session-token from process-data -->
    <!-- NOTE:This operation must be the last one in the BP -->

    <operation name="CONNECT:Direct Server Begin Session Service">
      <participant name="CDServerBPResponseServiceInst/>
      <output message="BPResponseParams">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="BPResponseResults" from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

Connect:Direct Server CopyFrom Service

The following table provides an overview of the Connect:Direct Server CopyFrom service:

System Name	Connect:Direct Server CopyFrom Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Direct
Description	This service copies a file from a remote Connect:Direct server. To ensure that the copy performed by this service can restart if a Application crash interrupts the copy, the business process (in which this service is used) must have the recovery level set to Auto-resume. If so marked, the business process and the copy are resumed automatically when Application restarts, if the remote Connect:Direct node is still active. When resumed, the copy is restarted from the beginning (or from the last checkpoint if configured).
Business usage	A business user would use this service to copy a file from a remote Connect:Direct.
Usage example	A trading partner copies a business document from a remote Connect: Direct outside the organization.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	<p>Whenever the Connect:Direct Server CopyFrom service is used in a business process, the Connect:Direct Server Primitive CopyFrom service is exercised behind the scenes as part of a system business process to perform the actual copy operation.</p> <p>The Connect:Direct Server CopyFrom service has tight integration with the Connect:Direct Server Begin Session service. A session must be established between two Connect:Direct nodes using the Begin Session service to execute the CopyFrom service successfully. The Connect:Direct Server End Session service must be used to end the session established between the Connect:Direct nodes after the Connect:Direct Server CopyFrom service.</p>
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No
Invocation	A user who has permission to perform this activity must start the business process that runs this service.

Business process
context considerations

LocalStripBlanks and LocalXlate are optional parameters. Use these parameters only when you want to control non-standard behavior and you have a complete understanding of your environment, the local content, the target requirement, data translation, and character set conversion.

If LocalStripBlanks and LocalXlate are not explicitly assigned in the business process (see example below), translation occurs automatically during transmission.

Example of a BP that does not explicitly assign LocalStripBlanks and LocalXlate parameters:

```
<operation name="CD Server Copy To Service">
  <participant name="ConnectDirectServerCopyToService001"/>
  <output message="CopyTo">
    <assign to="." from="*"></assign>
    <assign to="RemoteFileName">purchase-order.txt</assign>
  </output>
  <input message="inmsg">
    <assign to="CopyToResults" from="*"></assign>
  </input>
</operation>
```

If LocalStripBlanks and LocalXlate are explicitly assigned in the business process (see example below), the business process controls the transfer and the specified values are used.

Note: If the LocalXlate parameter is set to 'Yes', the SysOpts parameter must also specify 'xlate (yes)'.

Example of a BP that explicitly assigns LocalStripBlanks and LocalXlate parameters:

```
<operation name="CD Server Copy To Service">
  <participant name="ConnectDirectServerCopyToService001"/>
  <output message="CopyTo">
    <assign to="." from="*"></assign>
    <assign to="BinaryMode">NO</assign>
    <assign to="LocalXlate">NO</assign>
    <assign to="LocalStripBlanks">NO</assign>
    <assign to="RemoteFileName">purchase-order.txt</assign>
  </output>
  <input message="inmsg">
    <assign to="CopyToResults" from="*"></assign>
  </input>
</operation>
```

Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified below. <p>Advanced Errors</p> <ul style="list-style-type: none"> ◆ Invalid Document Error – The business process does not have a document to be copied to a target Connect:Direct. ◆ CopyFrom Remote File Error – An error related to the file on the remote Connect:Direct node. Possible reasons may be invalid file path or invalid file permissions. ◆ Invalid Sysopts Error – Sysopts might contain Invalid characters ◆ Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. ◆ Connect:Direct Server Error – A generic error associated with the Connect:Direct Server has occurred. ◆ Connect:Direct Error – A generic error associated with Connect:Direct has occurred. ◆ Connect:Direct Service Error – A generic error associated with this service has occurred.
Restrictions	None
Persistence level	None
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Server CopyFrom Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server CopyFrom service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Configure the pre-installed copy of the Connect:Direct Server CopyFrom service. For information, see *Configuring the Connect:Direct Server CopyFrom Service* on page 200.
3. Use the Connect:Direct Server CopyFrom service in a business process.

Configuring the Connect:Direct Server CopyFrom Service

To configure the Connect:Direct Server CopyFrom service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.

Field	Description
BinaryMode	<p>Specifies the mode for Connect:Direct File transfer. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Yes - simple binary mode ◆ No - (default) - text mode ◆ VB - a variable block binary transfer with Connect:Direct on an OS/390
CheckpointInterval	<p>Specifies the byte interval for checkpoint support, which allows restart of interrupted transmissions at the last valid checkpoint point and therefore reduces the time to retransmit the file. Optional. Valid value is number of bytes, in the format: <i>nnnnnnnnnn</i>. Application stores checkpoint information for 30 days, after which it is automatically purged.</p> <p>If your database is corrupted, purge this checkpoint information by performing a cold restart. To do this, edit the adapter configuration Server Start Option parameter to Cold Start. Once you have performed the cold start, edit the adapter configuration to change the setting back to Warm Start.</p> <p>You can also purge the checkpoint information from the database manually.</p>
Compression	<p>Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9. Increasing value indicates higher attempt at compression. Values greater than 2 may not result in further compression.</p>
LocalDocumentName	<p>Name of the document in Application to which the remote file should be copied. Optional.</p>
LocalFromCodePage	<p>Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:</p> <ul style="list-style-type: none"> ◆ Code page specified is supported by the local Java VM ◆ Parameter LocalToCodePage is defined ◆ Text mode file transfer (BinaryMode = No) <p>Optional. Valid value is the name of a codepage supported by the local Java VM.</p> <p>For example:</p> <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>
LocalStripBlanks	<p>Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default).</p>

Field	Description
LocalToCodePage	<p>Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:</p> <ul style="list-style-type: none"> ◆ Code page specified is supported by the local Java VM ◆ Parameter LocalFromCodePage is defined ◆ Text mode file transfer (BinaryMode = No) <p>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:</p> <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>
LocalXlate	<p>Specifies if the Connect:Direct Server adapter is to perform EBCDIC to ASCII translation before creating a new document from the received remote file. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No.</p>
RemoteFileName	<p>Name of the file that should be copied from the remote Connect:Direct server. Enter the legal characters in the file name. Required.</p>
SessionToken	<p>Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:</p> <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval
Sysopts	<p>Remote system operations to be implemented on the remote Connect:Direct node. Optional.</p> <p>For example (for copying to Connect:Direct Windows only):</p> <pre>Datatype(text) xlate(yes)</pre> <p>For more information, see your Connect:Direct documentation.</p>

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct CopyFrom service:

Field	Description
BinaryMode	Specifies the mode for Connect Direct File transfer. Optional. Valid values are: <ul style="list-style-type: none">◆ Yes - simple binary mode◆ No - (default) - text mode◆ VB - a variable block binary transfer with Connect:Direct on an OS/390
CheckpointInterval	Specifies the byte interval for checkpoint support, which enables restart of interrupted transmissions at the last valid checkpoint point and to reduce the time to retransmit the file. Optional. Valid value is number of bytes, in the format: <i>nnnnnnnnnn</i> .
Compression	Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9.
DCB	Specifies Data Control Block (DCB), a group of attributes specified as name-value pairs separated by commas. Used only on OS/390 platform. Optional. Valid attributes are: <ul style="list-style-type: none">◆ recordlength◆ blocksize◆ recordformat◆ datasetorganization <p>Note: More DCB attributes are available but only these are supported in Application. Any other attribute will cause the copy to fail.</p> <p>For more information, see the Connect:Direct OS/390 user documentation.</p>
LocalDocumentName	Name of the document in Application to which the remote file should be copied. Optional.
LocalFromCodePage	Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions: <ul style="list-style-type: none">◆ Code page specified is supported by the local Java VM◆ Parameter LocalToCodePage is defined◆ Text mode file transfer (BinaryMode = No) Optional. Valid value is the name of a codepage supported by the local Java VM. For example: <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>
LocalStripBlanks	Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default).

Field	Description
LocalToCodePage	<p>Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:</p> <ul style="list-style-type: none"> ◆ Code page specified is supported by the local Java VM ◆ Parameter LocalFromCodePage is defined ◆ Text mode file transfer (BinaryMode = No) <p>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:</p> <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>
LocalXlate	<p>Specifies if the Connect:Direct Server adapter is to perform EBCDIC to ASCII translation before creating a new document from the received remote file . This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No.</p>
RemoteFileName	<p>Name of the file that should be copied from the remote Connect:Direct server. The path to the file should also be included in the name, if required by the remote node. Required. Enter the legal characters in the file name.</p>
SessionToken	<p>Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:</p> <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval
Sysopts	<p>Remote system operations to be implemented on the remote Connect:Direct node. Optional.</p> <p>For example (for copying to Connect:Direct Windows only):</p> <pre>Datatype(text) xlate(yes)</pre> <p>For more information, see your Connect:Direct documentation.</p>

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Server CopyFrom service to the business process:

Parameter	Description
LocalCDNodeName	Local Connect:Direct server instance name. Required.
RemoteCDNodeName	Remote Connect:Direct server instance name. Required.
BytesReceived	Number of bytes received as part of copy operation. Required. Valid value is a positive integer.
Status	Indicates the status of the copy operation. Required. Valid values are SUCCESS and ERROR.
CopyStartDateTime	Indicates copy operation begin date and time. Required. Valid value is in the format: YYYY MM DD hhmm.
CopyEndDateTime	Indicates copy operation end date and time. Required. Valid value is in the format: YYYY MM DD hhmm.

Business Process Example

The following example illustrates using the Connect:Direct Server CopyFrom service in a business process. The business process copies a file from the remote Connect:Direct server, which is identified with a session token.

```
<process name=" CopyFromExample">

  <sequence name="CDServerCopyFromGIS">
    <!-- Begin Server Session -->
    <operation name="CONNECT:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">TEST1_CDSEVER</assign>
        <assign to="RemoteCDNodeName">REMOTECDNAME1</assign>
        <assign to="RemoteUserId">userid1</assign>
        <assign to="RemotePasswd">password1</assign>
      </output>
      <input message="inmsg">
        <assign to="BeginSessionResults" from="*"></assign>
      </input>
    </operation>

    <!-- Perform Copy (Pnode-INbound) -->
    <operation name="CONNECT:Direct Server Copy From Service">
      <participant name="CDServerCopyFrom"/>
      <output message="CopyFrom">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
          from="//BeginSessionResults/SessionToken/node()"></assign>
        <assign to="RemoteFileName">samplecopyfrom.txt</assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

    <assign to="LocalDocumentName">samplelocal</assign>
    <assign to="BinaryMode">no</assign>
    <assign to="Sysopts">datatype(text) xlate(yes)</assign>

</output>
<input message="inmsg">
  <assign to="CopyFromResults" from="*"></assign>
</input>
</operation>

<!-- End Server Session -->
<operation name="CONNECT:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="EndSessionResults" from="*"></assign>
  </input>
</operation>

</sequence>
</process>

```

Connect:Direct Server CopyTo Service

The following table provides an overview of the Connect:Direct Server CopyTo service:

System Name	Connect:Direct Server CopyTo Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect Direct
Description	This service copies a file to a remote Connect:Direct server. To confirm that the copy performed by this service can restart if a Application crash interrupts the copy operation, the business process (in which this service is used) must have the recovery level set to Auto-resume. If so marked, the business process and the copy are resumed automatically when Application restarts, if the remote Connect:Direct node is still active. When resumed, the copy is restarted from the last checkpoint.
Business usage	A business user would use this service to copy a file to a remote Connect:Direct.
Usage example	A trading partner copies a business document to a remote Connect:Direct server outside the organization.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	<p>Whenever the Connect:Direct Server CopyTo service is used in a business process, the Connect:Direct Server Primitive CopyTo service is exercised behind the scenes as part of a system business process to perform the actual copy operation.</p> <p>The Connect:Direct Server CopyTo service has tight integration with the Connect:Direct Server Begin Session service. A session must be established between two Connect:Direct nodes using the Begin Session service in order to execute the CopyTo service successfully. The Connect:Direct Server End Session service must be used to end the session established between the Connect:Direct nodes after the Connect:Direct Server CopyTo service.</p>
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No
Invocation	A user who has permission to perform this activity must start the business process that runs this service.

Business process
context considerations

LocalStripBlanks and LocalXlate are optional parameters. Use these parameters only when you want to control non-standard behavior and you have a complete understanding of your environment, the local content, the target requirement, data translation, and character set conversion.

If LocalStripBlanks and LocalXlate are not explicitly assigned in the business process (see example below), translation occurs automatically during transmission.

Example of a BP that does not explicitly assign LocalStripBlanks and LocalXlate parameters:

```
<operation name="CD Server Copy To Service">
  <participant name="ConnectDirectServerCopyToService001"/>
  <output message="CopyTo">
    <assign to="." from="*"></assign>
    <assign to="RemoteFileName">purchase-order.txt</assign>
  </output>
  <input message="inmsg">
    <assign to="CopyToResults" from="*"></assign>
  </input>
</operation>
```

If LocalStripBlanks and LocalXlate are explicitly assigned in the business process (see example below), the business process controls the transfer and the specified values are used.

Note: If the LocalXlate parameter is set to 'Yes', the SysOpts parameter must also specify 'xlate (yes)'.

Example of a BP that explicitly assigns LocalStripBlanks and LocalXlate parameters:

```
<operation name="CD Server Copy To Service">
  <participant name="ConnectDirectServerCopyToService001"/>
  <output message="CopyTo">
    <assign to="." from="*"></assign>
    <assign to="BinaryMode">NO</assign>
    <assign to="LocalXlate">NO</assign>
    <assign to="LocalStripBlanks">NO</assign>
    <assign to="RemoteFileName">purchase-order.txt</assign>
  </output>
  <input message="inmsg">
    <assign to="CopyToResults" from="*"></assign>
  </input>
</operation>
```

Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified below. <p>Advanced Errors</p> <ul style="list-style-type: none"> ◆ Invalid Document Error – Business process does not have document to be copied to a target Connect:Direct server. ◆ CopyTo Remote File Error – An error related to file on remote Connect:Direct server. Might the file error or file path is invalid or invalid file permissions. ◆ Invalid Sysopts Error – Sysopts might contain Invalid characters ◆ Connect:Direct Server Error – Generic error associated with the Connect:Direct server has occurred. ◆ Connect:Direct Error – Generic error associated with the Connect:Direct server has occurred. ◆ Connect:Direct Service Error – Generic error associated with this service has occurred.
Restrictions	None
Persistence level	None
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Server CopyTo Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server CopyTo service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Configure the pre-installed copy of the service. For information, see *Configuring the Connect:Direct Server CopyTo Service* on page 209.
3. Use the Connect:Direct Server CopyTo service in a business process.

Configuring the Connect:Direct Server CopyTo Service

To configure the Connect:Direct Server CopyTo service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.

Field	Description
BinaryMode	<p>Specifies the mode for Connect:Direct File transfer. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Yes - simple binary mode ◆ No - (default) - text mode ◆ VB - a variable block binary transfer with Connect:Direct on an OS/390
CheckpointInterval	<p>Specifies the byte interval for checkpoint support, which allows restart of interrupted transmissions at the last valid checkpoint point, reducing the time it takes to retransmit the file. Optional. Valid value is number of bytes, in the format: <i>nnnnnnnnnn</i>.</p> <p>Application stores checkpoint information for 30 days, after which it is automatically purged.</p> <p>If your database is corrupted, purge this checkpoint information by performing a cold restart. To do this, edit the adapter configuration Server Start Option parameter to Cold Start. Once you have performed the cold start, edit the adapter configuration to change the setting back to Warm Start.</p> <p>You can also purge the checkpoint information from the database manually.</p>
Compression	<p>Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9.</p>
DCB	<p>Specifies Data Control Block (DCB), a group of attributes specified as name-value pairs separated by commas. One, some, or all values may be included. Pattern is name=value with comma between, as in the following example:</p> <p style="padding-left: 40px;">recordlength=200, blocksize=0, recordformat=FB, datasetorganization=PS, unittype=cart, expirationdate=06365, volumecount=99</p> <p>Used only on OS/390 platform. Optional. Valid attributes are:</p> <ul style="list-style-type: none"> ◆ recordlength ◆ blocksize ◆ recordformat ◆ datasetorganization ◆ expirationdate ◆ volumecount ◆ unittype <p>Note: More DCB attributes are available but only these are supported in Application. Any other attribute will cause the copy to fail.</p> <p>For more information, see the Connect:Direct for OS/390 user documentation.</p>
Disposition	<p>Specifies how the file gets created on the remote Connect:Direct server. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ New – Specifies that the CopyTo service will create the destination file. The copy fails if the file exists ◆ rpl – Specifies that the destination file name replaces any existing file or, if none exists, allocates a new file

Field	Description
LocalDocumentId	A valid document ID for an existing document in Application. When this field is left blank, the primary document provided to the business process is used. Optional.
LocalFileNameLabel	Specifies a label for the name of the local file. This file name label is provided to the remote Connect:Direct node for informational purposes only. Optional.
LocalFromCodePage	<p>Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:</p> <ul style="list-style-type: none"> ◆ Code page specified is supported by the local Java VM ◆ Parameter LocalToCodePage is defined ◆ Text mode file transfer (BinaryMode =No) <p>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:</p> <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>
LocalStripBlanks	Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default).
LocalToCodePage	<p>Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:</p> <ul style="list-style-type: none"> ◆ Code page specified is supported by the local Java VM ◆ Parameter LocalFromCodePage is defined ◆ Text mode file transfer (BinaryMode =No) <p>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:</p> <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>
LocalXlate	Specifies if the Connect:Direct Server adapter is to perform ASCII to EBCDIC translation before copying a file to a remote Connect:Direct Server. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No.

Field	Description
CDRecordType	Identifies to the remote Connect:Direct which class to process the service. For example: 1 Optional. For more information, see the Connect:Direct for OS/390 user documentation.
RemoteFileName	Name that should be given to the file that is copied to the remote Connect:Direct server. Required.
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval
Sysopts	Remote system operations to be implemented on the remote Connect:Direct node. Optional. For example (for copying to Connect:Direct Windows only): Datatype(text) xlate(yes) For more information, see your Connect:Direct documentation.

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct CopyTo service:

Field	Description
BinaryMode	Specifies the mode for Connect Direct File transfer. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Yes - simple binary mode ◆ No - (default) - text mode ◆ VB - a variable block binary transfer with Connect:Direct on an OS/390
CheckpointInterval	Specifies the byte interval for checkpoint support, which enables restart of interrupted transmissions at the last valid checkpoint point, reducing the time it takes to retransmit the file. Optional. Valid value is number of bytes, in the format: <i>nnnnnnnnnn</i> .

Field	Description
Compression	Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9.
DCB	<p>Specifies Data Control Block (DCB), a group of attributes specified as name-value pairs separated by commas. One, some, or all values may be included. Pattern is name=value with comma between, as in the following example:</p> <p>recordlength=200, blocksize=0, recordformat=FB, datasetorganization=PS, unittype=cart, expirationdate=06365, volumecount=99</p> <p>Used only on OS/390 platform. Optional. Valid attributes are:</p> <ul style="list-style-type: none"> ◆ recordlength ◆ blocksize ◆ recordformat ◆ datasetorganization ◆ expirationdate ◆ volumecount ◆ unittype <p>Note: More DCB attributes are available but only these are supported in Application. Any other attribute will cause the copy to fail.</p> <p>For more information, see the Connect:Direct for OS/390 user documentation.</p>
Disposition	<p>Specifies how the file gets created on the remote Connect:Direct server. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ New – Specifies that the CopyTo service will create the destination file. The copy fails if the file exists ◆ rpl – Specifies that the destination file name replaces any existing file or, if none exists, allocates a new file
LocalDocumentId	A valid document ID for an existing document in Application. Optional.
LocalFileNameLabel	Specifies a label for the name of the local file. This file name label is provided to the remote Connect:Direct node for informational purposes only. Optional.
LocalFromCodePage	<p>Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:</p> <ul style="list-style-type: none"> ◆ Code page specified is supported by the local Java VM ◆ Parameter LocalToCodePage is defined ◆ Text mode file transfer (BinaryMode = No) <p>Optional. Valid value is the name of a codepage supported by the local Java VM.</p> <p>For example:</p> <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>

Field	Description
LocalStripBlanks	Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default).
LocalToCodePage	<p>Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:</p> <ul style="list-style-type: none"> ◆ Code page specified is supported by the local Java VM ◆ Parameter LocalFromCodePage is defined ◆ Text mode file transfer (BinaryMode = No) <p>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:</p> <pre><assign to="LocalFromCodePage">Cp1251</assign> <assign to="LocalToCodePage">Cp1047</assign></pre> <p>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings.</p>
LocalXlate	Specifies if the Connect:Direct Server adapter is to perform ASCII to EBCDIC translation before copying a file to a remote Connect:Direct Server. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No.
RemoteFileName	Name that should be given to the file that is copied to the remote Connect:Direct server. Required.
SessionToken	<p>Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:</p> <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval
Sysopts	<p>Remote system operations to be implemented on the remote Connect:Direct node. Optional. For example (for copying to Connect:Direct Windows only):</p> <pre>Datatype(text) xlate(yes)</pre> <p>For more information, see your Connect:Direct documentation.</p>

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Server CopyTo service to the business process:

Parameter	Description
LocalCDNodeName	Local Connect:Direct server instance name. Required.
RemoteCDNodeName	Remote Connect:Direct server instance name. Required.
BytesSent	Number of bytes sent as part of Copy operation. Required. Valid value is a positive integer.
Status	Indicates the status of the Copy operation. Required. Valid values are SUCCESS and ERROR.
CopyStartDateTime	Indicates Copy operation begin date and time. Required. Valid value is in the format: YYYY MM DD hhmm.
CopyEndDateTime	Indicates Copy operation end date and time. Required. Valid value is in the format: YYYY MM DD hhmm.

Business Process Example

The following example illustrates using the Connect:Direct Server CopyTo service in a business process. The business process copies the current primary document to the remote Connect:Direct server, which is identified with a session token.

```
<process name=" CopyToExample">

  <sequence name="CDServerCopyToGIS">
    <!-- Begin Server Session -->
    <operation name="CONNECT:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">TEST1_CDSEVER</assign>
        <assign to="RemoteCDNodeName">REMOTECDNODE1</assign>
        <assign to="RemoteUserId">user1</assign>
        <assign to="RemotePasswd">password1</assign>
      </output>
      <input message="inmsg">
        <assign to="BeginSessionResults" from="*"></assign>
      </input>
    </operation>

    <operation name="CONNECT:Direct Server Copy To Service">
      <participant name="CDServerCopyTo"/>
      <output message="CopyTo">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
          from="//BeginSessionResults/SessionToken/node()"></assign>
        <assign to="RemoteFileName">sample.txt</assign>
        <assign to="BinaryMode">no</assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

        <assign to="Sysopts">datatype(text) xlate(no)</assign>

    </output>
    <input message="inmsg">
        <assign to="CopyToResults" from="*"></assign>
    </input>
</operation>

<!-- End Server Session -->
<operation name="CONNECT:Direct Server End Session Service">
    <participant name="CDServerEndSession"/>
    <output message="EndSession">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
            from="//BeginSessionResults/SessionToken/node()"></assign>
    </output>
    <input message="inmsg">
        <assign to="EndSessionResults" from="*"></assign>
    </input>
</operation>

</sequence>
</process>

```

Connect:Direct Server End Session Service

The following table provides an overview of the Connect:Direct Server End Session service:

System Name	Connect:Direct Server End Session Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Direct
Description	This service is used to end a session established between two Connect:Direct nodes.
Business usage	A business user would use this service to terminate a session with a remote Connect:Direct Node. The service uses the Connect:Direct Server adapter as the communications mechanism for the outbound request to end the session.
Usage example	A Application business process initiates a session with a remote Connect:Direct node using the Connect:Direct Begin Session service and the Connect:Direct Server adapter. During the course of the session, business activity occurs: for example, a file is copied from the remote Connect:Direct node. After the operation is complete, the business process uses the Connect:Direct End Session service to terminate the session.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms
Related services	<ul style="list-style-type: none">◆ Connect:Direct Server Begin Session service – used to initiate the session that will be terminated by this service.◆ Connect:Direct Server adapter – The Connect:Direct End Session service uses a Connect:Direct Server adapter to communicate with a remote Connect:Direct server.
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No
Invocation	This service is run by a business process to terminate a session with a remote Connect:Direct node.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified below. ◆ Invalid Session Token – The Session Token specified to identify the Session was not valid. ◆ Connect:Direct Server Error – A generic error associated with the Connect:Direct Server has occurred. ◆ Connect:Direct Error – A generic error associated with Connect:Direct has occurred. ◆ Connect:Direct Service Error – A generic error associated with this service has occurred.
Restrictions	You must use this service with a Connect:Direct Begin Session service. The session token information in process data is necessary to configure the service.
Persistence level	None
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Server End Session Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server End Session service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Create a Connect:Direct Server End Session service configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Direct Server End Session service. For information, see *Configuring the Connect:Direct Server End Session Service* on page 218.
4. Use the Connect:Direct Server End Session service in a business process.

Configuring the Connect:Direct Server End Session Service

To configure the Connect:Direct Server End Session service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.

Field	Description
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Server End Session service:

Field	Description
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Server End Session service to the business process:

Parameter	Description
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none">◆ SessionId◆ ProcessNumber◆ ProcessName◆ SubmitDateTime◆ RemoteCDNodeName◆ RemoteUserId◆ RemotePasswd◆ PlexClass◆ MaxRestartRetries◆ RestartRetryInterval
SessionEndTime	Specifies the date time when the session was ended. Required.

Business Process Example

The following example illustrates using the Connect:Direct Server End Session service in a business process:

```
<process name="CDInterop_SessionBeginAndEnd">

  <sequence name="CDServerSessionBeginGIS">
    <!-- Begin Server Session -->
    <operation name="CONNECT:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">TEST1_CDSEVER</assign>
        <assign to="RemoteCDNodeName">NMAYAKUNTLAVM2K</assign>
        <assign to="RemoteUserId">ksticken</assign>
        <assign to="RemotePasswd">kirk123</assign>
      </output>
      <input message="inmsg">
        <assign to="BeginSessionResults" from="*"></assign>
      </input>
    </operation>

    <!-- End Server Session -->
    <operation name="Connect:Direct Server End Session Service">
      <participant name="CDServerEndSession"/>
    </operation>
  </sequence>
</process>
```

```
<output message="EndSession">
  <assign to="." from="*"></assign>
  <assign to="SessionToken"
    from="//BeginSessionResults/SessionToken/node()"></assign>
</output>
<input message="inmsg">
  <assign to="EndSessionResults" from="*"></assign>
</input>
</operation>

</sequence>
</process>
```

Connect:Direct Server Run Job Service

The following table provides an overview of the Connect:Direct Server Run Job service:

System Name	Connect:Direct Server Run Job Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect Direct
Description	This service submits work to the host operating system that executes asynchronously to the remaining steps in a process. The work can execute on either the local or remote node.
Business usage	To execute a job on a mainframe, write a business process that uses the Run Job service to execute a script on a remote Connect:Direct node on OS/390.
Usage example	Executing a script asynchronously on a remote Connect:Direct server that compresses and archives log files.
Preconfigured?	No. A Connect:Direct Server adapter must be configured before using this service.
Requires third party files?	No
Platform availability	All supported platforms
Related services	The Connect:Direct Server Run Job service is designed to work surrounded by a Connect:Direct Server Begin Session service and a Connect:Direct Server End Session service. It uses the Connect:Direct Server adapter to communicate with remote Connect:Direct servers.
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No
Invocation	The business process running this service must be started by a user with permission to issue the Run Job statement on the target Connect:Direct server.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – Normal execution with the return parameters specified below.◆ File not found error – Specified file was not found.◆ Access to file denied error – Specified file could not be accessed.◆ Unable to start file error – Specified file could not be started.

Restrictions	Connect:Direct nodes on iSeries do not support System Options (Sysopts), an optional parameter for this service.
Persistence level	None
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Server Run Job Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server Run Job service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Create a Connect:Direct Server Run Job service configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Direct Server Run Job service. For information, see *Configuring the Connect:Direct Server Run Job Service*.
4. Use the Connect:Direct Server Run Job service in a business process.

Configuring the Connect:Direct Server Run Job Service

To configure the Connect:Direct Server Run Job service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.
SessionToken	Specifies the identifier for the session established between two Connect:Direct server nodes. Required.
Sysopts	Specifies the parameters appropriate for the operation being performed. Required. Valid values are: <ul style="list-style-type: none"> ◆ <code>pgm(filespec)</code> specifies which .EXE or .BAT file to run. ◆ <code>cmd(command parms)</code> specifies a system command and any arguments that this command requires. ◆ <code>args(arguments)</code> specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify <code>pgm</code>.

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Server Run Job service:

Parameter	Description
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none">◆ SessionId◆ ProcessNumber◆ ProcessName◆ SubmitDateTime◆ RemoteCDNodeName◆ RemoteUserId◆ RemotePasswd◆ PlexClass◆ MaxRestartRetries◆ RestartRetryInterval
Sysopts	Specifies the parameters appropriate for the operation being performed. Required. Valid values are: <ul style="list-style-type: none">◆ <i>pgm(filespec)</i> specifies which .EXE or .BAT file to run.◆ <i>cmd(command parms)</i> specifies a system command and any arguments that this command requires.◆ <i>args(arguments)</i> specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify <i>pgm</i>.

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Server Run Job service to the business process:

Parameter	Description
LocalCDNodeName	The Application Connect:Direct Server adapter name.
RemoteCDNodeName	The remote Connect:Direct server IP address.
Status	Indicates whether or not the remote job was executed. Note: This does not provide address the success or failure of the executed job, only that it was executed.

Business Process Examples

The following examples illustrate using the Connect:Direct Server Run Job service in a UNIX environment and a Windows environment:

The following is a UNIX example:

```
<process name="RunJobServiceExample">
  <sequence>
    <operation name="CONNECT:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">CDSERVER_ADAPTER_TEST1</assign>
        <assign to="RemoteCDNodeName">CDSERVER_ADAPTER_TEST2</assign>
        <assign to="RemoteUserId">USERID</assign>
        <assign to="RemotePasswd">PASSWORD</assign>
        <assign to="PlexClass">ABC</assign>
      </output>
      <input message="inmsg">
        <assign to="BeginSessionResults" from="*"></assign>
      </input>
    </operation>
    <operation name="Connect:Direct Server Run Job Service">
      <participant name="CDServerRunJob"/>
      <output message="RunJob">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
          from="//BeginSessionResults/SessionToken/node()"></assign>
        <assign to="Sysopts" from="'cd /home/sterlingcommerce/script; sh
          runScript.sh argument1'"></assign>
      </output>
      <input message="inmsg">
        <assign to="RunJob_Results" from="*"></assign>
      </input>
    </operation>
    <operation name="CONNECT:Direct Server End Session Service">
      <participant name="CDServerEndSession"/>
      <output message="EndSession">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
          from="//BeginSessionResults/SessionToken/node()"></assign>
      </output>
      <input message="inmsg">
        <assign to="EndSessionResults" from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

The following is a Windows example:

```
<process name="RunJobServiceExample">
  <sequence>
    <operation name="CONNECT:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
```

```

    <assign to="." from="*"></assign>
    <assign to="LocalCDNodeName">CDSERVER_ADAPTER_TEST1</assign>
    <assign to="RemoteCDNodeName">CDSERVER_ADAPTER_TEST2</assign>
    <assign to="RemoteUserId">USERID</assign>
    <assign to="RemotePasswd">PASSWORD</assign>
    <assign to="PlexClass">ABC</assign>
  </output>
  <input message="inmsg">
    <assign to="BeginSessionResults" from="*"></assign>
  </input>
</operation>
<operation name="Connect:Direct Server Run Job Service">
  <participant name="CDServerRunJob"/>
  <output message="RunJob">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
    <assign to="Sysopts"
      from="'pgm(D:\SterlingProject\CDInterop\Demo\test.bat)
      args(TEST)'"></assign>
  </output>
  <input message="inmsg">
    <assign to="RunJob_Results" from="*"></assign>
  </input>
</operation>
<operation name="CONNECT:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="EndSessionResults" from="*"></assign>
  </input>
</operation>
</sequence>
</process>

```

Connect:Direct Server Run Task Service

The following table provides an overview of the Connect:Direct Server Run Task service:

System Name	Connect:Direct Server Run Task Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect Direct
Description	This service executes a program or command synchronously to the remaining steps in a Process. The program or command can execute locally or on the remote Connect:Direct node. User programs can be run in different environments.
Business usage	You want to execute a job on a mainframe. You can write a business process that uses the Run Task service to execute a script on a remote Connect:Direct node on OS/390 and then wait for the completion of the run task operation.
Usage example	You execute a script that compresses and archives log files synchronously on a remote Connect:Direct Server.
Preconfigured?	A Connect:Direct Server adapter must be configured before using this service
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	The Connect:Direct Server Run Task Service works between a Connect:Direct Server Begin Session Service and a Connect:Direct Server End Session service,
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No
Invocation	The business process running this service must be started by a user with permission to issue the Run Task statement on the target Connect:Direct server.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – Normal execution with the return parameters specified below.◆ File not found Error – Specified file was not found.◆ Access to file denied Error – Specified file could not be accessed.◆ Unable to start file Error – Specified file could not be started.◆ Program Failure Error – Failure of the program to be executed by the specified file.

Restrictions	Connect:Direct nodes on iSeries do not support System Options (Sysopts), an optional parameter for this service.
Persistence level	None
Testing considerations	None

Implementing the Connect:Direct Server Run Task Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server Run Task service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Create a Connect:Direct Server Run Task service configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Direct Server Run Task. For information, see *Configuring the Connect:Direct Server Run Task Service* on page 228.
4. Use the Connect:Direct Server Run Task service in a business process.

Configuring the Connect:Direct Server Run Task Service

To configure the Connect:Direct Server Run Task service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.
Program	<p>Specifies the parameters appropriate for the operation being performed. Required.</p> <p>Note: In Application Release 3.1, the name of this parameter was System Options. The valid values depend on the destination Connect:Direct server. For example, the valid values for Connect:Direct Windows are:</p> <ul style="list-style-type: none"> ◆ <code>pgm</code> (<i>filespec</i>) specifies which .exe or .bat file to run. ◆ <code>cmd</code> (<i>command parms</i>) specifies a system command and any arguments that this command requires. ◆ <code>args</code> (<i>arguments</i>) specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify <code>pgm</code>.

Field	Description
SessionToken	<p>Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:</p> <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval
SystemOpts	<p>Specifies the arguments passed to the program when it is started. Optional.</p> <p>Note: Connect:Direct AS/400 nodes do not support SystemOpts.</p>

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Server Run Task service:

Field	Description
Program	<p>Specifies the parameters appropriate for the operation being performed. Required.</p> <p>Note: In Application Release 3.1, the name of this parameter was System Options.</p> <p>The valid values depend on the destination Connect:Direct server. For example, the valid values for Connect:Direct Windows are:</p> <ul style="list-style-type: none"> ◆ <i>pgm (filespec)</i> specifies which .exe or .bat file to run. ◆ <i>cmd (command parms)</i> specifies a system command and any arguments that this command requires. ◆ <i>args (arguments)</i> specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify <i>pgm</i>.

Field	Description
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval
SystemOpts	Specifies the arguments passed to the program when it is started. Optional. Note: Connect:Direct AS/400 nodes do not support SystemOpts.

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Connect:Direct Server Run Task service to the business process:

Parameter	Description
LocalCDNodeName	The Application Connect:Direct Server adapter name
RemoteCDNodeName	The remote Connect:Direct server IP address
Status	Indicates whether or not the remote job was executed. Note: This does not address the success or failure of the executed job, only that it was executed.

Business Process Example for Connect:Direct for UNIX

Each parameter's value and syntax is defined by the remote Connect:Direct platform and underlying operating system. When the remote platform operating system is UNIX/Linux, the command and its arguments, if any, are specified as the Program parameter value and in the same syntax that they would be entered at a shell prompt. For example, to execute the 'ls' command with the '-l' option (single column output) and send the output to a file called 'ls.out' locally on a UNIX/Linux host, type the following at a shell prompt:

```
$ ls -l > ls.out
```

To specify this operation in the RunTask Service, assign the text to the RunTask Service Program parameter in the business process:

```
<assign to="Program">ls -l /tmp &gt; ls.out</assign>
```

Note: The redirection operator (>) must be encoded (>) to prevent XML processing errors.

The following is a business process for UNIX with the Program parameter:

```
<process name = "SampleRunTask">
  <sequence>
    <operation name="CD Server Begin Session Service">
  ...
  </operation>
  <operation name="CD Server Run Task Service">
    <participant name="CDServerRunTask"/>
    <output message="CDServerRunTaskServiceTypeInputMessage">
      <assign to="Program">ls -l /tmp &gt; ls.out</assign>
  <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
  <operation name="CD Server End Session Service">
  ...
  </operation>
</sequence>
</process>
```

Business Process Example for Connect:Direct for Windows

When a Connect:Direct Windows node receives a request to execute a task, the business process must specify whether or not the task requires a cmd shell to execute. Connect:Direct Windows defines a syntax for communicating this information. To execute a program, the pgm and args keywords are used:

```
<assign to="Program">pgm(\path\file.exe) args(arg1 arg2 arg3)</assign>
```

Note: The args keyword is only used when arguments are needed. The path is computed according to Connect:Direct rules. If the Program directory is defined for the submitting user on the remote node, the pgm value will be appended to its value prior to use.

To execute a Windows command:

```
<assign to="Program">cmd(command arg1 arg2 arg3)</assign>
```

The following business process demonstrates how to configure the Connect:Direct RunTask Service:

```
<process name = "SampleRunTask">
  <sequence>
    <operation name="CD Server Begin Session Service">
  ...
  </operation>
  <operation name="CD Server Run Task Service">
    <participant name="CDServerRunTask"/>
    <output message="CDServerRunTaskServiceTypeInputMessage">
  <assign to="Program">cmd(dir \tmp &gt; \tmp\dir.out)</assign>
</output>
    <input message="inmsg">
```

```
        <assign to="." from="*"></assign>
    </input>
</operation>
<operation name="CD Server End Session Service">
...
    </operation>
</sequence>
</process>
```

Parameter for Connect:Direct for Z/OS

The SystemOpts parameter value is appended to the Program parameter value. Use this parameter when the remote node is Connect:Direct for Z/OS. In this case, the Program parameter value specifies the remote program name only. The remote program parameters, if any, are specified using the SystemOpts parameter.

Connect:Direct Server Submit Service

The following table provides an overview of the Connect:Direct Server Submit service:

System Name	Connect:Direct Server Submit Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect Direct
Description	This service is used to submit a Connect:Direct process to another Connect:Direct node. The submitted process must reside in a file on the destination Connect:Direct node.
Business usage	A business user would use this service to initiate work on a remote Connect:Direct node. This service performs the equivalent of a Submit statement in a Connect:Direct process. During execution of a Connect:Direct process, the Submit statement causes another Connect:Direct process to be submitted to either the Pnode, which is the node with process control, or the Snode, which is the remote node that participates in the process. Any parameter values that are specified in the Submit statement override the parameter values contained in a process statement that is named by the file parameter.
Usage example	A business user has a file that exists on a remote node where Connect:Direct is running. This file is needed on another node. The user creates a business process that submits a Connect:Direct process to a remote Connect:Direct server to transfer the file from one node to the other.
Preconfigured?	A Connect:Direct Server adapter must be configured before using this service.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Connect:Direct Server adapter
Application requirements	<ul style="list-style-type: none">◆ Perimeter server◆ Connect:Direct Windows 4.1.00 or later◆ Connect:Direct UNIX 3.5.00 or later◆ Connect:Direct OS/390 4.4.00 or later◆ Connect:Direct OS/400 3.5.00 or later
Initiates business processes?	No
Invocation	The business process running this service must be started by a user with permission to issue the Submit statement on the target Connect:Direct server.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ Invalid Business Process Error – User tried to submit an invalid business process in Application. ◆ User Permission Error – User does not have permission to submit the business process in Application. ◆ Connect:Direct Server Error – A generic error associated with the Connect:Direct Server has occurred. ◆ Connect:Direct Error – A generic error associated with Connect:Direct has occurred. ◆ Connect:Direct Service Error – A generic error associated with the Connect:Direct service has occurred.
Restrictions	If the Submit Process request is issued to a remote Connect:Direct node that is operating on OS/400, the request will fail. The OS/400 Connect:Direct node does not support Submit Process.
Persistence level	None
Testing considerations	Debug information for this service can be found in the Connect:Direct Interop log files.

Implementing the Connect:Direct Server Submit Service

Before implementing any specific Connect:Direct Server services, outline the business tasks to be completed between Connect:Direct and Application, and decide which Connect:Direct Server services are needed.

To implement the Connect:Direct Server Submit service, complete the following tasks:

1. Configure the Connect:Direct Server adapter to be used with this service. For information, see *Connect:Direct Server Adapter*.
2. Create a Connect:Direct Server Submit service configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Direct Server Submit. For information, see *Configuring the Connect:Direct Server Submit Service* on page 234.
4. Use the Connect:Direct Server Submit service in a business process.

Configuring the Connect:Direct Server Submit Service

To configure the Connect:Direct Server Submit service, you must define the fields in Application and in the GPM.

Field	Description
Config	Name of the service configuration. Required.
RemoteProcessFile	Process file on the remote Connect:Direct server. This is the process that is submitted. Required.
RemoteProcessParms	The parameters or the arguments that can be passed dynamically to the process file on remote Connect:Direct server. Optional. UNIX Example: SNODE=10.20.80.117;1364

Field	Description
SubmitUserId	User ID for remote system other than the remoteUserId specified in the Connect:Direct Server Begin Session service. Optional. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service).
SubmitPasswd	Password for remote system for user ID specified in SubmitUserId. Required when SubmitUserId is specified. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses password associated with the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service)
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval

Parameters Passed from Business Process to Service

The following table describes the parameters passed from the business process to the Connect:Direct Server Submit service:

Parameter	Description
RemoteProcessFile	Process file on the remote Connect:Direct server. This is the process that is submitted. Required.
RemoteProcessParms	The parameters or the arguments that can be passed dynamically to the process file on remote Connect:Direct server. Optional. UNIX Example: SNODE=10.20.80.117;1364
SubmitUserId	User ID for remote system other than the remoteUserId specified in the Connect:Direct Server Begin Session service. Optional. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service).

Parameter	Description
SubmitPasswd	Password for remote system for user ID specified in SubmitUserId. Required when SubmitUserId is specified. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses password associated with the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service)
SessionToken	Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <ul style="list-style-type: none"> ◆ SessionId ◆ ProcessNumber ◆ ProcessName ◆ SubmitDateTime ◆ RemoteCDNodeName ◆ RemoteUserId ◆ RemotePasswd ◆ PlexClass ◆ MaxRestartRetries ◆ RestartRetryInterval

Parameters Passed From Service to Business Process

The following table describes the parameters passed from the Connect:Direct Server Submit service to the business process:

Parameter	Description
LocalCDNodeName	Specifies the local Connect:Direct which submitted the process. Required.
RemoteCDNodeName	Specified the remote Connect:Direct on which the process was submitted. Required.
Status	Specifies the status of the submit statement execution. Required. Valid values are SUCCESS and ERROR.
SubmittedProcessNum	Specifies the process number assigned to the process. Optionally returned (if > -1).
SubmittedProcessDateTime	Specifies the date and time when the submit statement is executed. Required.

Business Process Example

The following example illustrates using the Connect:Direct Server Submit service in a business process:

```
<process name=" SubmitStatementExample">
  <sequence name="CDServerSubmit">
    <!-- Begin Server Session -->
      <operation name="CONNECT:Direct Server Begin Session Service">
```

```

<participant name="CDServerBeginSession"/>
<output message="BeginSession">
  <assign to="." from="*"></assign>
  <assign to="LocalCDNodeName">TEST1_CDSEVER</assign>
  <assign to="RemoteCDNodeName">NMAYAKUNTLAVM2K</assign>
  <assign to="RemoteUserId">joeuser</assign>
  <assign to="RemotePasswd">joel23</assign>

</output>
<input message="inmsg">
  <assign to="BeginSessionResults" from="*"></assign>
</input>
</operation>

<operation name="Connect:Direct Server Submit Service">
  <participant name="CDServerSubmit"/>
  <output message="SubmitRequest">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
    <assign to="RemoteProcessFile">Lalitha.cdp</assign>
    <assign to="RemoteProcessParms">SNODE=00.00.00.000;1364</assign>
  </output>
  <input message="inmsg">
    <assign to="SubmitResults" from="*"></assign>
  </input>
</operation>

<!-- End Server Session -->
<operation name="CONNECT:Direct Server End Session Service">
  <participant name="CDServerEndSession"/>
  <output message="EndSession">
    <assign to="." from="*"></assign>
    <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
  </output>
  <input message="inmsg">
    <assign to="EndSessionResults" from="*"></assign>
  </input>
</operation>
</sequence>
</process>

```

Connect:Enterprise UNIX Server Adapter

The following table provides an overview of the Connect:Enterprise UNIX Server adapter:

System name	Connect:Enterprise UNIX Server Adapter
Graphical Process Modeler (GPM) category	This adapter does not appear in the GPM because it is not used directly in business processes.
Description	<p>This adapter works as a conduit between Connect:Enterprise UNIX and Application. It functions as a business process protocol daemon within Connect:Enterprise UNIX.</p> <p>When enabled, this adapter has a persistent connection to Connect:Enterprise UNIX. It can respond to a schedule notification from Connect:Enterprise UNIX (also called an auto connect) or it can respond to a Application service that initiates an unsolicited session (also called a remote connect).</p> <p>In the case of auto connect, Connect:Enterprise UNIX notifies the adapter when data (which meets the criteria of the schedule definition) is placed in a mailbox. The adapter:</p> <ul style="list-style-type: none">◆ Retrieves batch summary data and sets an acknowledgement flag (V flag) in the Connect:Enterprise UNIX system◆ Initiates the business process specified in the Connect:Enterprise UNIX schedule definition◆ Logs the notification event in Connect:Enterprise UNIX◆ Puts the Application business process ID into Connect:Enterprise UNIX. It also adds a hyperlink which can open a new browser window with the Application business process Detail Page. <p>In the case of a remote connect, the adapter runs by one or more of the Connect:Enterprise UNIX Server services.</p>
Business usage	Use this adapter for all communications with Connect:Enterprise UNIX. Used in conjunction with the Connect:Enterprise UNIX Server services, this adapter can extract or add data from a mailbox, submit queries, change process flags and log messages in Connect:Enterprise UNIX.
Usage example	A trading partner adds a batch to a Connect:Enterprise UNIX mailbox that is configured to initiate a Application business process. This adapter receives notification about the added batch and initiates the specified business process. The extract service in the business process runs the adapter to copy the batch from the mailbox and pass the batch into Application for additional processing.
Preconfigured?	Yes. One configuration called CEU Server Adapter BP is installed with Application. To use this configuration you must supply a valid Connect:Enterprise UNIX host and port, ID and password, and then you must enable the configuration.
Requires third party files?	No
Platform availability	All supported Application platforms

Related services	<p>This adapter is used by the following services:</p> <ul style="list-style-type: none"> ◆ <i>Connect:Enterprise UNIX Server Add Service</i> ◆ <i>Connect:Enterprise UNIX Server Extract Service</i> ◆ <i>Connect:Enterprise UNIX Server Begin Session Service</i> ◆ <i>Connect:Enterprise UNIX Server End Session Service</i> ◆ <i>Connect:Enterprise UNIX Server Batch Status Service</i> ◆ <i>Connect:Enterprise UNIX Server Log Service</i>
Application requirements	An active instance of Application and an active instance of Connect:Enterprise UNIX V2.2 or later is required.
Initiates business processes?	This adapter initiates a business process when it receives notification from Connect:Enterprise UNIX. The business process is specified in the Connect:Enterprise UNIX schedule definition.
Invocation	No
Business process context considerations	<p>When initiating a business process due to a Connect:Enterprise UNIX schedule, the business process context contains the AcdName (that is, the Auto Connect Definition name), SessionId and several other important pieces of information.</p> <p>For more information, see <i>Process Data</i> on page 241.</p>
Returned status values	None
Restrictions	None
Persistence level	None
Testing considerations	<p>When the adapter configuration is enabled, it means it is connected to Connect:Enterprise UNIX. Disabled means not connected to Connect:Enterprise UNIX.</p> <p>If Connect:Enterprise UNIX shuts down, the Connect:Enterprise UNIX Server adapter is disabled. However, when Connect:Enterprise UNIX comes back up, the adapter re-enables automatically.</p> <p>For more information, see <i>Testing the Connect:Enterprise UNIX Adapter</i> on page 247.</p>

System Requirements

For a complete list of requirements, see the *Connect:Enterprise UNIX Installation and Administration Guide*.

Licensing Requirements

You need appropriate adapter licensing in order to use one or more configurations of the Connect:Enterprise UNIX Server adapter.

How the Connect:Enterprise UNIX Server Adapter Works

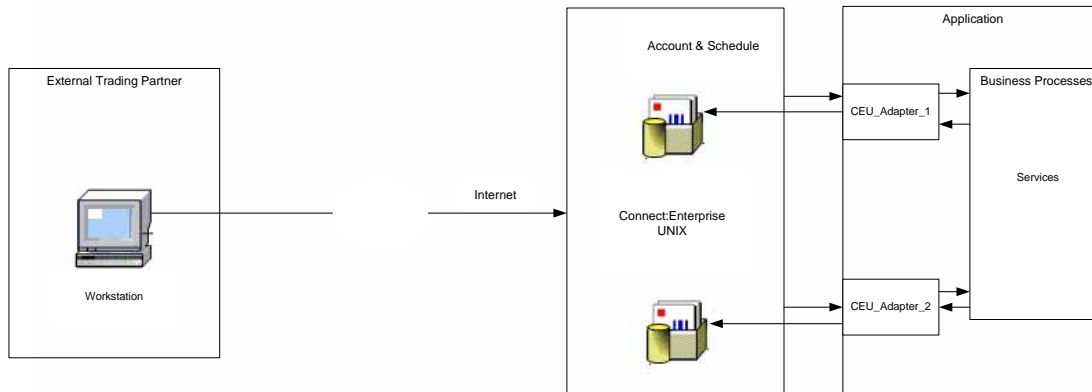
The following table shows what tasks are necessary to establish Application and Connect:Enterprise UNIX for interoperability:

Configuration Task	Application
Define an account (RSD—remote site definitions) to allow Application to connect into Connect:Enterprise UNIX. Note: This task is optional because the system is delivered with a predefined account called samp_bp.	Connect:Enterprise UNIX
Define a schedule (ACD—auto connect definitions) to indicate when to notify Application. Note: This task is optional because the system is delivered with a predefined schedule called samp_bp.acd	Connect:Enterprise UNIX
Configure the Connect:Enterprise UNIX Server adapter. Note: Though the preconfigured adapter, CEU Server Adapter BP, is delivered with the system, you must supply some additional parameters and enable the configuration	Application
Test connectivity between Application and Connect:Enterprise UNIX: 1 Enable the adapter 2 Perform cmusession quick test.	Application and Connect:Enterprise UNIX
Run the Interoperability Verification Demonstration.	Application and Connect:Enterprise UNIX

Interoperability

The Connect:Enterprise UNIX Server adapter is a Application component that acts as a conduit for connectivity with Connect:Enterprise UNIX and enables interoperability between Application and Connect:Enterprise UNIX.

The following figure shows how Application and Connect:Enterprise UNIX work together to exchange information. It also illustrates load balancing where multiple adapters connect to a single Connect:Enterprise UNIX host and port.



The adapter must be configured with a valid Connect:Enterprise UNIX host, port, ID, and password. After the adapter is configured and enabled, Connect:Enterprise UNIX registers the adapter as a Business Process protocol daemon.

The following describes how the Connect:Enterprise UNIX Server adapter responds to a notification schedule:

1. A configuration of the adapter receives notification from Connect:Enterprise UNIX when data meeting the criteria of a schedule notification is placed in a Connect:Enterprise UNIX mailbox.
2. The adapter retrieves summary data about the batch from the Connect:Enterprise UNIX mailbox according to parameters defined in the schedule.

For the summary data to be retrieved, the R (requestable) processing flag must be present. However, batch summary data with the following processing flags are not retrieved:

- ◆ P (Transmission in progress)
 - ◆ I (Incomplete)
 - ◆ D (Deleted)
 - ◆ T (Transmitted)
 - ◆ V (Application acknowledged)
3. The adapter initiates the Application business process (defined in the Connect:Enterprise UNIX schedule) and places the summary data into process data.
 4. The adapter waits for further instructions from the Application business process.

Process Data

In the case of an auto connect, the batch summary data identifies the adapter that was used, the Connect:Enterprise UNIX SessionId that has been started, and Auto Connect Definition (ACD) information

including specific batch summary data. At this point the business document is not in process data. The next step depends upon your business process.

In the case of remote connect, the content of the information placed in process data by the adapter varies depending upon the service that begins the business process.

CEUAdapterInstanceName and SessionId

The configuration name of the Connect:Enterprise UNIX Server adapter (CEUAdapterInstanceName) and SessionId are at the top level of the process data for visibility and use by the services in the business process, as in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CEUAdapterInstanceName>BP</CEUAdapterInstanceName>
  <SessionId>2057</SessionId>
  <ACD_INFO>
    .
    .
    .
  </ACD_INFO>
</ProcessData>
```

These parameters are important because they indicate which Connect:Enterprise UNIX Server adapter configuration started the business process and what session the services will be acting on behalf of. Because you can have multiple configurations of the adapter, each Connect:Enterprise UNIX Server service used in the business process needs to determine which adapter configuration to communicate with so that the business process activity is passed through the correct (requesting) adapter.

Note: Because the SessionId is put into process data during a schedule notification, it serves as an implied Begin Session; therefore, you do not need to use a Begin Session service in a business processes initiated from a schedule notification. However, you must still use an End Session service to bracket all activities within the same SessionId.

Note: The CEUAdapterInstanceName in process data overrides whatever adapter the services are configured to use. Therefore, if you configure a new Connect:Enterprise UNIX Server adapter you do not need to create a new suite of service configurations to point to the new adapter. The original Connect:Enterprise UNIX Server service configurations are reusable.

Business Process Parameters Defined in Connect:Enterprise UNIX

The business process parameters in the following example include mailbox information and additional user-defined XML content that is pertinent to the business process. The parameters are entered in Connect:Enterprise UNIX as “Additional Business Process Parameters” when the remote BP (Business Process protocol daemon) block is defined within a schedule.

```
<BusinessProcessParameters format="xml">
  <TP_INFO>
    <Mailboxes>
      <Rejected>bpreject</Rejected>
      <InProgress>bpinproc</InProgress>
      <Invoice>bpinvoic</Invoice>
      <Messages>bpmsg</Messages>
    </Mailboxes>
    <EmailAddress>jane_doe@myco.com</EmailAddress>
  </TP_INFO>
</BusinessProcessParameters>
```

```
</TP_INFO>
</BusinessProcessParameters>
```

Batch Summary Data

Other important pieces of information are the CEUBatchNumber and CEUBatchDateTime. These pieces of information are used when data is extracted. In the following example, you can see the batch summary data for two batches that were named *.po and located in the bporders mailbox. This information is batch summary data only and does not include the actual business document at this point in the business process.

```
.
.
.
<Batches>
  <Batch>
    <CEUBatchNumber>5262</CEUBatchNumber>
    <CEUMailboxId>Process Dataorders</CEUMailboxId>
    <CEUBatchId>CarOrder.po</CEUBatchId>
    <CEUBatchSize>208</CEUBatchSize>
    <CEUBatchDateTime>12/17/03 3:38 PM</CEUBatchDateTime>
    <CEUProcessFlagStatus>C,R,J,Z</CEUProcessFlagStatus>
  </Batch> <Batch>
    <CEUBatchNumber>5263</CEUBatchNumber>
    <CEUMailbox>bporders</CEUMailbox>
    <CEUBatchId>CarOrder.po</CEUBatchId>
    <CEUBatchSize>208</CEUBatchSize>
    <CEUBatchDateTime>12/17/03 3:40 PM</CEUBatchDateTime>
  </Batch>
</Batches>
</ACD_INFO>
</ProcessData>
```

Connect:Enterprise UNIX Server Services

When the Application business process starts, it uses the Connect:Enterprise UNIX Server services to perform activities on batches in or from the Connect:Enterprise UNIX mailbox.

These services, described below, are Application components that work in collaboration with the Connect:Enterprise UNIX Server adapter:

Connect:Enterprise UNIX Server Add service - Adds a batch to a Connect:Enterprise UNIX mailbox

Connect:Enterprise UNIX Server Extract service - Extracts a batch from a Connect:Enterprise UNIX mailbox

Connect:Enterprise UNIX Server Begin Session service - Begins a session with an instance of Connect:Enterprise UNIX

Connect:Enterprise UNIX Server End Session service - Ends a session with an instance of Connect:Enterprise UNIX

Connect:Enterprise UNIX Server Batch Status service - Queries and updates batch summary data. This service can reach into the Connect:Enterprise UNIX mailbox to find and even change data including status flags (process and data format), CEUMailboxId and CEUBatchId.

Connect:Enterprise UNIX Server Log service: Writes log messages for success or error conditions to make them visible in the Connect:Enterprise UNIX interface.

Note: If a business process ran by Connect:Enterprise UNIX is not active, and the Connect:Enterprise UNIX Server End session service was not used, the auto connect session (seen in cmusession) times out after 30 seconds.

Interoperability Tracking

Activities between Application and Connect:Enterprise UNIX can be tracked from either system.

The Connect:Enterprise UNIX Server adapter writes Application correlation records to enable searches for business processes in Application that contain a particular Connect:Enterprise UNIX AcdName or a SessionId.

The adapter also logs the notification event in Connect:Enterprise UNIX and identifies the Application business process ID that started. This information is visible in the Connect:Enterprise UNIX Detail Account and Schedule reports. The business process ID appears as a hyperlink. Clicking on the business process ID opens a browser window to the Detail page of that business process.

Although a configuration of the Connect:Enterprise UNIX Server adapter is included in Application, you can create a new configuration of the adapter.

Before creating a new configuration of the adapter, review the following tips:

You can use the Connect:Enterprise UNIX Server adapter named “BP” for your production environment. However, if you plan to use more than one configuration of the adapter with a single installation of Connect:Enterprise UNIX, you must create additional configurations with different names.

The Connect:Enterprise UNIX schedule that starts a session with Application points to a specific resource. This resource is the registered configuration of the Connect:Enterprise UNIX Server adapter. Therefore, it may be useful to create and name configurations of the adapter according to how they will be used in Application (for example: SAP, Rosetta, or UCCnet), and then have these multiple adapters connected to a single installation of Connect:Enterprise UNIX.

The following information is needed before you can complete configuration and enable a configuration of the Connect:Enterprise UNIX Server adapter:

Connect:Enterprise UNIX Host - The host name or IP address of the Connect:Enterprise UNIX instance that the adapter will connect to.

Connect:Enterprise UNIX Port - The port number the adapter will use to connect to the Connect:Enterprise UNIX application. This should be the main Connect:Enterprise UNIX port.

Connect:Enterprise UNIX User ID - The user ID required to access Connect:Enterprise UNIX. This is a Connect:Enterprise UNIX remote account.

Connect:Enterprise UNIX Password - The password required to access Connect:Enterprise UNIX.

Implementing the Connect:Enterprise UNIX Server Adapter

To implement the Connect:Enterprise UNIX Server adapter, complete the following tasks:

1. Activate your license for the Connect:Enterprise UNIX Server adapter. For information, see *Managing Services and Adapters*.

2. Create a Connect:Enterprise UNIX Server adapter configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Enterprise UNIX Server adapter. For information, see *Configuring the Connect:Enterprise UNIX Server Adapter* on page 245.
4. Test the Connect:Enterprise UNIX Server adapter. For information, see *Testing the Connect:Enterprise UNIX Adapter* on page 247.
5. Create configurations of any Connect:Enterprise UNIX Server adapter services needed for use with this adapter. See *Related services* on page 239 in the Overview table for a list of these services and where to find more information about each.
6. Use the Connect:Enterprise UNIX Server adapter and any additional configurations in a business process.

Configuring the Connect:Enterprise UNIX Server Adapter

To configure the Connect:Enterprise UNIX Server adapter, you must specify settings for the following fields in Application:

Field	Description
Name	<p>Unique and meaningful name for the adapter configuration. Required.</p> <p>Note: When enabled, the adapter name registers in Connect:Enterprise UNIX as a Business Process protocol daemon. When you perform a <code>cmusession</code> command in Connect:Enterprise UNIX, this name should be in the list of registered daemons.</p>
Description	<p>Meaningful description for the adapter configuration, for reference purposes. Required.</p>
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
CEU Host	<p>Host name or IP address of the Connect:Enterprise UNIX installation that the adapter will connect to.</p>
CEU Port	<p>Port number the adapter will use to connect to the Connect:Enterprise UNIX application.</p>
CEU User ID	<p>User ID required to access the Connect:Enterprise UNIX application when a business process must use a remote connection to Connect:Enterprise UNIX.</p>
CEU Password	<p>Password required to access the Connect:Enterprise UNIX application when a business process must use a remote connection to Connect:Enterprise UNIX.</p>

Field	Description
Maximum active CEU sessions	Maximum number of sessions that can be active at any given time. The minimum value is 1.
Listen Port	Port number on which Application will listen for a Connect:Enterprise UNIX Schedule notification. Valid value is a unique port number on the Application host machine that will not conflict with another process.

Caution: If the new configuration of the Connect:Enterprise UNIX Server adapter is not enabled, Application cannot communicate with Connect:Enterprise UNIX.

Output from Adapter to Business Process

The example below shows what the process data looks like when the Connect:Enterprise UNIX Server adapter runs a business process as a result of a Connect:Enterprise UNIX schedule notification:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
<CEUServerAdapterInstanceName>BP</CEUServerAdapterInstanceName>
  <SessionId>2057</SessionId>
  <ACD_INFO>
    <AcdName>samp_bp.acd</AcdName>
    <SessionId>2057</SessionId>
    <AdapterName>BP</AdapterName>
    <MailboxList>bporders</MailboxList>
    <BatchIdPattern>*.xml</BatchIdPattern>
    <PutOptions>$$CODE=A</PutOptions>
    <BusinessProcessParameters format="xml">
      <TP_INFO>
        <Mailboxes>
          <Rejected>bpreject</Rejected>
          <InProgress>bpinproc</InProgress>
          <Invoice>bpinvoic</Invoice>
          <Messages>bpmsg</Messages>
        </Mailboxes>
        <EmailAddress>john.doe@myco.com</EmailAddress>
      </TP_INFO>
    </BusinessProcessParameters>
  <Batches>
    <Batch>
      <CEUBatchNumber>9218</CEUBatchNumber>
      <CEUBatchDateTime>1/19/04 9:40 AM</CEUBatchDateTime>
      <CEUBatchSize>208</CEUBatchSize>
      <CEUMailboxId>bporders</CEUMailboxId>
      <CEUBatchId>PO.xml</CEUBatchId>
      <CEUProcessFlagStatus>C,R,J,Y</CEUProcessFlagStatus>
    </Batch>
    <Batch>
      <CEUBatchNumber>9219</CEUBatchNumber>
      <CEUBatchDateTime>1/19/04 9:41 AM</CEUBatchDateTime>
      <CEUBatchSize>208</CEUBatchSize>
      <CEUMailboxId>bporders</CEUMailboxId>
      <CEUBatchId>PO.xml</CEUBatchId>
      <CEUProcessFlagStatus>C,R,J,Y</CEUProcessFlagStatus>
    </Batch>
  </Batches>
</ProcessData>
```

```
</Batch>
</Batches>
</ACD_INFO>
</ProcessData>
```

High Concurrency Load Balancing

For scheduled notifications, all communications between Application and Connect:Enterprise UNIX use the adapter configuration in the auto connect request. Throughout the initiated business process, all services use the adapter that began the auto connection.

For remote connection, however, you can create a group of Connect:Enterprise UNIX Server adapters that can be used for load balancing. These adapters should be identically configured except for Name and Listen port. The group name is used for the CEUAdapterInstanceName parameter. When a business process uses an adapter from that group for the first service, the same adapter is used to manage all subsequent activity within the same session. At each new session the least recently used adapter in the group is used.

Testing the Connect:Enterprise UNIX Adapter

If Connect:Enterprise UNIX shuts down while it is connected to Application through the adapter, it will automatically reconnect when Connect:Enterprise UNIX restarts.

To verify that Application and Connect:Enterprise UNIX are communicating:

1. Configure the Connect:Enterprise UNIX Server adapter and save it. If it enables, you are connected.
2. To verify the connection in Connect:Enterprise UNIX, enter one of the following cmusession commands to find a registered configuration of the adapter.

To list only registered business process protocols:

```
cmusession -u <username> -p <password> -g
```

To list all registered protocols:

```
cmusession -u <username> -p <password>
```

If the Connect:Enterprise UNIX adapter name does not appear in the list when you perform the cmusession command, the most frequent problems encountered are:

- ◆ Invalid host IP address or port number
 - ◆ The adapter was not enabled when configured
3. If problems continue, turn on the special ceuinterop.log logging from the Operations menu, **Operations > System > Log**. All services and the adapter log to this file.

In addition, error statements are logged in the appserver log file (jboss.log or weblogic.log).

Connect:Enterprise UNIX Server Add Service

The following table provides an overview of the Connect:Enterprise UNIX Server Add service:

System name	Connect:Enterprise UNIX Server Add Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX
Description	This service works with the Connect:Enterprise UNIX Server adapter to add a batch to a Connect:Enterprise UNIX mailbox.
Business usage	<p>If used in a business process that is initiated from a Connect:Enterprise UNIX schedule notification, this service adds the batch and logs the activity into Connect:Enterprise UNIX as part of the schedule notification session.</p> <p>If the Add service is used in business process that begins an unsolicited session, the activity is represented in Connect:Enterprise UNIX as remote business process account activity.</p>
Usage example	<p>You want to send a business document to a trading partner who has a Connect:Enterprise UNIX mailbox.</p> <p>For more information, see <i>How the Connect:Enterprise UNIX Server Add Service Works</i> on page 249.</p>
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Extract service, Batch Status service, Log service.
Application requirements	An active instance of Application and Connect:Enterprise UNIX v2.2 or later are required for this service to work.
Initiates business processes?	No
Invocation	Runs as part of a business process.

Business process context considerations	<p>The service gets the Connect:Enterprise UNIX Server adapter name from one of two places:</p> <ul style="list-style-type: none"> ◆ when run in response to a schedule notification the CEUServerAdapterInstanceName is automatically placed in the process data ◆ when run by some other means, this parameters could come from a previous service, or it must be defined in this service's configuration. <p>If the process data does not already contain the CEUSessionId, then this service uses the Connect:Enterprise UNIX-generated CEUSessionId.</p> <p>Included in process data (when run by a schedule notification) are Put Options including data format and process flags. The Add service can carry these Put Options forward to be used when the batch is added, or you can override these flags by specifying these parameters in the Add service.</p>
Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified. ◆ Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. ◆ Invalid Session – Indicates an invalid session. ◆ Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. ◆ Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise Unix Server Adapter. ◆ CEU Service Error – A generic error associated with the service has occurred.
Restrictions	None
Persistence level	All
Testing considerations	<ol style="list-style-type: none"> 1 Successfully establish connection from Application to Connect:Enterprise UNIX through a Connect:Enterprise UNIX Server adapter. 2 Create a business process with the Add service. 3 Pass in a primary document and it should be added into Connect:Enterprise UNIX as remote connect activity. <p>For more information, see the predefined business process, CEUInterop-DemoPOAdd.</p>

How the Connect:Enterprise UNIX Server Add Service Works

The following describes what happens when this service is used to add a business document to a Connect:Enterprise UNIX mailbox as the result of an unsolicited remote connect.

1. The Application business process initiates a communication session with Connect:Enterprise UNIX through the Connect:Enterprise UNIX Server adapter. The adapter submits a user ID and password to gain access.
2. The adapter returns a CEUSessionId to the Begin Session service. The CEUSessionId is put in the process data.
3. Application adds a batch to a specific mailbox using the Connect:Enterprise UNIX Server Add service. This service works through the Connect:Enterprise UNIX Server adapter to accomplish the add. Included in the transfer are: CEUMailboxId, CEUBatchId, and the file being transferred.

4. Connect:Enterprise UNIX stores the data in the mailbox and returns a notification to the adapter indicating that the data was transferred successfully. The notification includes the batch number Connect:Enterprise UNIX assigned to the file and the creation date.
5. A flag ("J" flag) is set on the batch in Connect:Enterprise UNIX to indicate that Application added the batch.
6. Application sets the correlations between the business process that initiated the request and batch summary data from Connect:Enterprise UNIX. These include: CEUBatchId, CEUMailboxId, CEUBatchNumber, schedule definition name and CEUSessionId. These correlations allow subsequent searches for this information using the Advanced Search function.

Implementing the Connect:Enterprise UNIX Server Add Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Application, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Add service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. Configure the pre-installed copy of the service. For information, see *Configuring the Connect:Enterprise UNIX Server Add Service* on page 250.
3. Use the Connect:Enterprise UNIX Server adapter in a business process.

Configuring the Connect:Enterprise UNIX Server Add Service

To configure the Connect:Enterprise UNIX Server Add service, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Add service:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
CEU Server Adapter Instance Name (CEUServerAdapterInstanceName)	<p>The instance of the Connect:Enterprise UNIX adapter that the Add service should use. The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP.</p>

GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Add service in the GPM:

Field	Description
Config	Name of the service configuration. Required.
CEUBatchId	<p>The name to be assigned to the batch when it is placed in the Connect:Enterprise UNIX repository.</p> <p>The primary document name is used if no name is specified.</p>
CEUDataFormat	<p>Sets the data format flag: ASCII, BINARY, EBCDIC.</p> <p>In an auto connect scenario, Connect:Enterprise UNIX supplies this flag in the process data within the schedule definition: ACD_INFO/PutOptions.</p> <p>In a remote connect scenario, Connect:Enterprise UNIX can use the Put Option defined in the remote account (RSD).</p> <p>Optional.</p> <ul style="list-style-type: none"> ◆ If not specified, Connect:Enterprise UNIX uses the Put Option defined in the schedule (ACD_INFO) or in the remote account (RSD). ◆ Default - C (Collected), J (BP protocol flag), Y (binary data format) ◆ If specified, overrides the Connect:Enterprise UNIX Put Option.
CEUMailboxId	<p>The ID of the Connect:Enterprise UNIX mailbox that is to receive the document. The document is added to the mailbox for the account name if no CEUMailboxId is specified.</p>

Field	Description
CEUProcessOption	<p>Sets the extract and transmit process option flag. Valid values are:</p> <ul style="list-style-type: none"> ◆ Extract never, transmit once (N flag) PutOptions; TO=YES CEUProcessOption; "Unextractable" ◆ Extract once, transmit never (U flag) PutOptions; EO=YES CEUProcessOption; "Nontransmittable" ◆ Extract repeatedly, transmit never Has no effect ◆ Extract repeatedly, transmit once (R flag) PutOptions; XMIT=YES CEUProcessOption; "Requestable" ◆ Extract repeatedly, transmit repeatedly (M flag) PutOptions; MULTXMIT=YES CEUProcessOption; "Multi-transmit" <p>In an auto connect scenario, Connect:Enterprise UNIX supplies this flag in the process data within the schedule definition: ACD_INFO/PutOptions.</p> <p>In a remote connect scenario, Connect:Enterprise UNIX can use the Put Option defined in the remote account (RSD).</p> <p>Optional.</p> <ul style="list-style-type: none"> ◆ If not specified, Connect:Enterprise UNIX uses the Put Option defined in the schedule (ACD_INFO) or in the remote account (RSD). ◆ Default - C (Collected), J (BP protocol flag), Y (binary data format) ◆ If specified, overrides the Connect:Enterprise UNIX Put Option.
CEUServerAdapterInstanceName	Normally, this is already defined.
CEUTriggerAutoConnect	Indicates that a schedule session should be triggered when the batch is added to the Connect:Enterprise UNIX mailbox.
GISDocumentId	If the primary document is not the document to be added, then identify the Application document ID to be added as a Connect:Enterprise UNIX batch.

Output from Service to Business Process

The following table describes the information that is placed in process data by the Connect:Enterprise UNIX Server Add service (Message From Service):

Parameter	Description
CEUBatchNumber	The Unique batch identifier within Connect:Enterprise UNIX. In conjunction with the creation date, batches can be uniquely identified within Connect:Enterprise UNIX.
CEUBatchDateTime	The creation date of the batch. Format: 'yyyyMMdd'T'hmm'

Parameter	Description
CEUMailboxId	The Connect:Enterprise UNIX Mailboxes to which the document was added.
CEUBatchId	The name to be assigned to the new Connect:Enterprise UNIX batch.
CEUProcessFlagStatus	The process flags as reported by Connect:Enterprise UNIX after the batch is added to the mailbox. C (Collected), R, J (Added by Application), and Y (binary)
GISDocumentId	The Application document ID added to the Connect:Enterprise UNIX mailbox.
CEUBatchSize	Size of batch in bytes.
CEUTriggerAutoconnect	Indicates if a schedule session is triggered when the batch is added to the mailbox.
CEUSessionId	The Connect:Enterprise UNIX session ID associated with this activity.

Business Process Example

The following example illustrates a small part of the BPML that utilizes the Connect:Enterprise UNIX Server Add service.

```
<!-- Add Msg to Acme mailbox -->
  <operation name="CEU Server Add Service Add Service">
    <participant name="CEUServerAdd"/>
      <output message="AddRequest">
        >
          <assign to="CEUMailboxId">Invoice</assign>
          <assign to="CEUBatchId">MessageName</assign>
          <assign to="CEUTriggerAutoConnect">Yes</assign>
        </output>
      <input message="inmsg">
        <assign to="MessagefromService" from="*"></assign>
      </input>
    </operation>
```

Connect:Enterprise UNIX Server Batch Status Service

The following table provides an overview of the Connect:Enterprise UNIX Server Batch Status service:

System name	Connect:Enterprise UNIX Server Batch Status Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX
Description	This service passes queries and updates to Connect:Enterprise UNIX through the Connect:Enterprise UNIX Server adapter.
Business usage	This service is multifunctional. Use it to: <ul style="list-style-type: none">◆ retrieve summary data for one or more batches◆ determine the state of process status flags and data format status flags for one or more batches◆ change any of the status flags or change a mailbox ID or a batch ID.
Usage example	<p>A batch exists in a mailbox but is not extractable by Application because of the process flag settings. For example, the batch is allowed only one extract and it has already been extracted. This service is used to update the process flag on the batch so that it can be extracted again.</p> <p>For more information, see <i>How the Connect:Enterprise UNIX Server Batch Status Service Works</i> on page 255.</p>
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Add service, Extract service, Log service.
Application requirements	Active instances of Application and Connect:Enterprise UNIX v2.2 or later are required for this service to work.
Initiates business processes?	No
Invocation	Runs as part of a business process.

Business process context considerations	<p>The result of a Batch Status service query is a series of data within a STATUS_INFO tag. This service functions as a query only unless specific updates are defined in the service configuration.</p> <p>When this service is used in a business process that is initiated by a schedule notification from Connect:Enterprise UNIX:</p> <ul style="list-style-type: none"> ◆ It can use the CEUAdapterInstanceName and CEUSessionID that is provided automatically in the process data ◆ A looping business process is normally used to process each element in the auto connect definition ACD_INFO/Batches/Batch. (This information is in the process data.)
Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified. ◆ Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. ◆ Invalid Session – Indicates an invalid session. ◆ Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. ◆ Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise Unix Server Adapter. ◆ CEU Service Error – A generic error associated with the service has occurred.
Restrictions	None
Persistence level	All
Testing considerations	For information, see the business process, CEUInterop_DemoPOProcess

How the Connect:Enterprise UNIX Server Batch Status Service Works

The Connect:Enterprise UNIX Server Batch Status service sends a query to Connect:Enterprise UNIX and, optionally, can be used to update process flags, the CEUBatchId or the CEUMailboxId.

The following example describes what happens when a business process is initiated by a Connect:Enterprise UNIX schedule notification, but then, as a result of a downstream activity's error, the Batch Status service is used to alter the Connect:Enterprise UNIX transmitted (T) flag so that the batch can be extracted again.

1. The Connect:Enterprise UNIX Server adapter initiates a business process in response to a schedule notification from Connect:Enterprise UNIX.
2. The business process runs the extract service which copies the batch from the mailbox and sets the transmitted flag on the batch indicating that Application extracted the batch. In this case the batch is allowed to be transmitted only once.
3. The business process runs a translation of the batch from EDI to XML. However, a map error occurs during translation and the batch is not translated properly.
4. The error triggers the start of the Batch Status service.

- The Batch Status service passes a message to Connect:Enterprise UNIX to remove the transmitted flag from the batch. The STATUS_INFO returned by Connect:Enterprise UNIX is placed in process data.

Here is an example of output from a Batch Status service that was used to update *all* batches in the bporders mailbox by turning the T flag off. Notice that one batch was returned.

```
<STATUS_INFO>
  <QueryParms>
    <CEUMailboxId>bporders</CEUBatchNumber>
  </QueryParms>
  <UpdateParms>
    <CEUProcessFlagUpdate>!T</CEUProcessFlagUpdate>
  </UpdateParms>
  <Batches>
    <Batch>
      <CEUBatchNumber>189</CEUBatchNumber>
      <CEUBatchDateTime>2/21/04 8:24PM</CEUBatchDateTime>
      <CEUBatchSize>123456967</CEUBatchSize>
      <CEUMailboxId>bporders</CEUMailboxId>
      <CEUBatchId>PriDoc.128M</CEUBatchId>
      <CEUProcessFlagStatus>A,R,Z</CEUProcessFlagStatus>
    </Batch>
  </Batches>
</STATUS_INFO>
```

- Now the batch can be extracted again.

The following example illustrates the XPath statements in the subsequent Extract service. These statements derive the needed information from the STATUS_INFO content in the process data.

```
<!-- Extract batch by xpath to the provided batch number and datetime in ProcessData.
-->

<operation name="CEU Extract Service">
<participant name="CEUServerExtract"/>
  <output message="ExtractRequest">
    <assign to="CEUBatchNumber"
from="STATUS_INFO/Batches/Batch/CEUBatchNumber/text()">
    </assign>
    <assign to="CEUBatchDateTime"
from="STATUS_INFO/Batches/Batch/CEUBatchDateTime/text()">
    </assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
```


Implementing the Connect:Enterprise UNIX Server Batch Status Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Application, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Batch Status service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server Batch Status service. For information, see *Configuring the Connect:Enterprise UNIX Server Batch Status Service* on page 257.
3. Use the Connect:Enterprise UNIX Server Batch Status service in a business process.

Configuring the Connect:Enterprise UNIX Server Batch Status Service

To configure the Connect:Enterprise UNIX Server Batch Status service, you can optionally specify field settings in Application, but you must specify field settings in the GPM.

Application Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Batch Status service:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default)◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. For more information about service groups see <i>Managing Services and Adapters</i> .
CEU Server Adapter Instance Name	The instance of the Connect:Enterprise UNIX adapter that the Batch Status service should use.
(CEUServerAdapterInstanceName)	The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP.

GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Batch Status service in the GPM:

Field	Description
Config	Name of the service configuration. Required.
CEUBatchFromDate	Specifies all batches that were created on or after this date. Optional.
CEUBatchId	Specifies the name of one batch in Connect:Enterprise UNIX. Optional.
CEUBatchNumber	Specifies the batch number. Optional.
CEUBatchOriginator	Specifies the originator of the batch. Optional.
CEUBatchToDate	Specifies all batches that were created on or before this date. Optional.
CEUMailboxId	Specifies the mailbox identification where the batch resides. Optional.
CEUProcessFlagQuery	Lists the process flags that you are sending the query about. This is a comma-delimited string of process flags. Use an exclamation point (!) before a flag to indicate a lack of the flag. Optional. Example: Y, R, T, P, !V would mean to query batches that are binary (Y), requestable (R), transmitted (T) or transmission in progress (P) but which have not been acknowledged by Application (!V)
CEUProcessFlagUpdate	Specifies what the update to the flags should be. Needs the same format as in CEUProcessFlagQuery. Optional. Example: V, !R, would mean to change the queried batches to acknowledged (!V) and not requestable (!R).
CEUServiceAdapterInstanceName	The instance of the Connect:Enterprise UNIX adapter that the Batch Status service should use. This is typically already defined. Required.
NewCEUBatchId	Specifies the identification you want to use to update the CEUBatchId. Optional.
NewCEUMailboxId	Specifies the identification you want to use to update the CEUMailboxId. Optional.

Note: The service is used as a query only, unless one or more of these three parameters are defined: CEUProcessFlagUpdate, NewCEUBatchId, NewCEUMailboxId.

Output from Service to Business Process

The following table describes the information placed in process data by the Connect:Enterprise UNIX Server Batch Status service (Message From Service).

Parameter	Description
CEUBatchNumber	This is the batch number provided to this service as input. (Combined with CEUBatchDateTime this uniquely identifies the batch.)
CEUBatchId	This is the name of the batch that was provided to the service as input prior to any update.

Parameter	Description
CEUMailboxId	This is the name of the mailbox that was provided to the service as input prior to any update.
CEUProcessFlagQuery	These are the process flags that were queried by the service.
CEUProcessFlagUpdate	These are the process flags that were updated by the Batch Status service.
CEUBatchDateTime	The batch date and time. Combined with CEUBatchNumber, this uniquely identifies the batch queried or changed by the service.
CEUBatchSize	Size of the message in bytes
CEUProcessFlagStatus	This is the status of all process flags as they appear after the Batch Status service.

Business Process Example

The following example illustrates a small part of the BPML that utilizes the Connect:Enterprise UNIX Server Batch Status service.

```
<operation name="CEUServerBatchStatusServiceType">
  <participant name="CEUServerBatchStatus"/>
  <output message="CEUServerBatchStatusServiceTypeInputMessage">
    <assign
to="CEUBatchNumber" from="ACD_INFO/Batches/Batch[number(//numBatches/text())]/CEUBatc
hNumber/text()" >
      </assign>
      <assign to="." from="*">
      </assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
```

Connect:Enterprise UNIX Server Begin Session Service

The following table provides an overview of the Connect:Enterprise UNIX Server Begin Session service:

System name	Connect:Enterprise UNIX Server Begin Session Service
Graphical Process Modeler (GPM) categories	All Services, Application > Sterling Commerce > Connect:Enterprise UNIX
Description	This service passes a begin session request to the Connect:Enterprise UNIX Server adapter. This service works with the adapter to initiate a session with Connect:Enterprise UNIX and populate the Application process data with a CEUSessionId.
Business usage	Use this service in a business process that initiates an unsolicited session to Connect:Enterprise UNIX. This type of session is called a remote connect. In a remote connect, use this service with the End Session service to group intervening activities in a single Connect:Enterprise UNIX session.
Usage example	You want to start a session with Connect:Enterprise UNIX in order to retrieve a group of documents, translate them from EDI to XML and then put them back into Connect:Enterprise UNIX mailboxes. In addition, you want all the activities run by an instance of this business process to be tracked by a single session ID. For more information, see <i>How the Connect:Enterprise UNIX Server Begin Session Service Works</i> on page 261.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; End Session service, Extract Session service, Add service, Batch Status service, Log service.
Application requirements	An active instance of Application and Connect:Enterprise UNIX v2.2 or later are required for this service to work.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	This service is not needed when a Connect:Enterprise UNIX schedule initiates the business process. In this case there is an implied Begin Session because the Connect:Enterprise UNIX supplies the CEUSessionId. In a remote connect scenario, if the Begin Session service is not used, each Connect:Enterprise UNIX Server service, such as extract and add, runs with its own CEUSessionId

Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified. ◆ Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. ◆ Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. ◆ Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise Unix Server Adapter. ◆ CEU Service Error – A generic error associated with the service has occurred.
Restrictions	None
Persistence level	All
Testing considerations	For information, see the predefined business process, CEUInterop_DemoPOAddNoTrigger

How the Connect:Enterprise UNIX Server Begin Session Service Works

The following steps describe what happens when the Connect:Enterprise UNIX Server Begin Session service is used in a business process that initiates a remote connect.

1. The Application business process attempts to initiate a communication session with Connect:Enterprise UNIX through the Connect:Enterprise UNIX Server adapter. The adapter submits a user ID and password to gain access.
2. The adapter returns a CEUSessionId to the Begin Session service. The CEUSessionId is put in the process data.
3. All Connect:Enterprise UNIX Server services (add, extract, batch status and log) that follow the Begin Session service and precede the End Session service are captured under that one CEUSessionId.

Implementing the Connect:Enterprise UNIX Server Begin Session Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Application, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Begin Session service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server Begin Session service. For information, see *Configuring the Connect:Enterprise UNIX Server Begin Session Service* on page 262.
3. Use the Connect:Enterprise UNIX Server Begin Session service in a business process.

Configuring the Connect:Enterprise UNIX Server Begin Session Service

To configure the Connect:Enterprise UNIX Server Begin Session service, you must specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default)◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. For more information about service groups see <i>Managing Services and Adapters</i> .
CEU Server Adapter Instance Name (CEUServerAdapterInstanceName)	The instance of the Connect:Enterprise UNIX adapter that the Begin Session service should use. The preconfigured Begin Session service uses the preconfigured adapter named CEU Server Adapter BP.

Output from Service to Business Process

The following table describes the output from the Connect:Enterprise UNIX Server Begin Session service to the business process:

Parameter	Description
CEUSessionId	Specifies the identifier for the session established between the adapter and the Connect:Enterprise UNIX instance.

Business Process Example

The following example illustrates a part of the BPML that uses a Connect:Enterprise UNIX Server Begin Session service to run a remote connect session. All activity is bracketed by the Begin Session and End Session services. The Begin Session service sets CEUSessionId in process data. The Add and Extract services use that CEUSessionId, and End Session removes the CEUSessionId from process data.

```
<process name="CEUInterop_AddExtractService">
  <sequence>
    <operation name="CEUServerBeginSessionServiceType">
```

```

<participant name="CEU Server Begin Session" />
  <output message="CEU Server Begin Session Service Type Input Message">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="CEU Add Service">
  <participant name="Test CEU Server Add" />
  <output message="Add Request">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="CEU Add Service Results" from="*"></assign>
  </input>
</operation>
<!-- Remove Primary document to verify extract service -->
<operation name="Release Service">
  <participant name="Release Service" />
  <output message="Release Service Type Input Message">
    <assign to="TARGET" from="'Primary Document'"></assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="CEU Extract Service">
  <participant name="Test CEU Server Extract" />
  <output message="Extract Request">
    <assign to="CEU Batch Number"
from="CEU Add Service Results/CEU Batch Number/text()"></assign>
  </output>
  <input message="inmsg">
    <assign to="CEU Extract Service Results" from="*"></assign>
  </input>
</operation>
<operation name="CEU Server End Session Service Type">
  <participant name="CEU Server End Session" />
  <output message="CEU Server End Session Service Type Input Message">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</process>

```

Connect:Enterprise UNIX Server End Session Service

The following table provides an overview of the Connect:Enterprise UNIX Server End Session service:

System name	Connect:Enterprise UNIX Server End Session Service
Graphical Process Modeler (GPM) categories	All Services, Application > Sterling Commerce > Connect:Enterprise UNIX
Description	This service works with the Connect:Enterprise UNIX Server adapter to notify Connect:Enterprise UNIX that the session is over. This service also removes the CEUSessionId from the Application process data.
Business usage	Use this service in a business process to end an auto connect session to Connect:Enterprise UNIX. Note that Connect:Enterprise UNIX holds resources for the session until it is ended. In a business process that initiates a remote connect session, use this service in conjunction with the Begin Session service to group multiple activities within one CEUSessionId.
Usage example	You want all activities performed in a business process to be tracked, correlated, and reported as a single Connect:Enterprise UNIX session.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, Extract Session service, Add service, Batch Status service, Log service.
Application requirements	An active instance of Application and Connect:Enterprise UNIX v2.2 or later are required for this service to work.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	Always use this service in business processes initiated by a Connect:Enterprise UNIX schedule notification. Use of this service with the Begin Session service in a business process that initiates a remote connect is optional. For more information, see <i>How the Connect:Enterprise UNIX Server End Session Service Works</i> on page 265.

Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified. ◆ Invalid Session - Indicates an invalid session. ◆ Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. ◆ Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. ◆ Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise UNIX Server Adapter. ◆ CEU Service Error – A generic error associated with the service has occurred.
------------------------	--

Restrictions	None
--------------	------

Persistence level	All
-------------------	-----

Testing considerations	For information, see the predefined business process, CEUInterop_DemoPOAddNoTrigger
------------------------	---

How the Connect:Enterprise UNIX Server End Session Service Works

In the case of Auto Connect:

When the business process is initiated by a Connect:Enterprise UNIX schedule notification, place the End Session service after the final Connect:Enterprise UNIX Server service in the business process.

If the End Session service is not included in the business process, the session with Connect:Enterprise UNIX will end automatically 30 seconds after the business process becomes inactive.

Non-Connect:Enterprise UNIX activities, such as translations, can be placed before the End Session service. This is useful, for example, if you want the Connect:Enterprise UNIX Server Log service to notify Connect:Enterprise UNIX if errors result from the other activities.

In the case of Remote Connect:

When the business process initiates a remote connect session with Connect:Enterprise UNIX, you can use the Begin Session and End Session services to group activities under the same CEUSessionId.

If the Begin/End Session services are not used, each Connect:Enterprise UNIX Server service, such as Add or Extract, runs with its own CEUSessionId. Each activity is represented in Connect:Enterprise UNIX reporting as separate, unrelated entries.

Implementing the Connect:Enterprise UNIX Server End Session Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Application, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server End Session service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.

2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server End Session service. For information, see *Configuring the Connect:Enterprise UNIX Server End Session Service* on page 266.
3. Use the Connect:Enterprise UNIX Server End Session service in a business process.

Configuring the Connect:Enterprise UNIX Server End Session Service

To configure the Connect:Enterprise UNIX Server End Session service, you must specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
CEU Server Adapter Instance Name (CEUServerAdapterInstanceName)	<p>The instance of the Connect:Enterprise UNIX adapter that the End Session service should use.</p> <p>The preconfigured End Session service uses the preconfigured adapter named CEU Server Adapter BP.</p>

Output from Business Process to Service

The following table describes the output from the business process to the Connect:Enterprise UNIX Server End Session service:

Parameter	Description
CEUSessionId	Specifies the identifier for the session established between the adapter and the Connect:Enterprise UNIX instance.

Business Process Example

A Business Process protocol daemon that runs by a schedule notification and runs synchronously where all activity is under the CEUSessionId that is found in process data:

```
<process name="CEUInterop_ACDEExtractLoop">
```

```

<rule name="notFinished">
  <condition>number(counter/text()) > 0</condition>
</rule>
<sequence name="Sequence Start">
  <assign to="counter"
from="count(//ACD_INFO/Batches/Batch/BatchNumber/text())"></assign>
<sequence name="theLoop">
  <!-- Remove Primary document to verify extract service -->
  <operation name="Release Service">
    <participant name="ReleaseService"/>
    <output message="ReleaseServiceTypeInputMessage">
      <assign to="TARGET" from="'PrimaryDocument'"></assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
<operation name="CEU Extract Service">
  <participant name="TestCEUServerExtract"/>
  <output message="ExtractRequest">
    <assign to="CEUBatchNumber"
from="ACD_INFO/Batches/Batch[number(//counter/text())]/BatchNumber/text()">
    </assign>
  </output>
  <input message="inmsg">
    <assign to="MessageFromExtractService" from="*" append="true"/>
    <assign to="." from="PrimaryDocument" append="true"/>
  </input>
  </operation>
<assign to="counter" from="number(counter/text()) - 1"></assign>
  <choice>
    <select>
      <case ref="notFinished" activity="doRepeat"/>
    </select>
  </choice>
<repeat name="doRepeat" ref="theLoop"/>
</choice>
</sequence>
<operation name="CEUServerEndSessionServiceType">
  <participant name="CEUServerEndSession"/>
  <output message="CEUServerEndSessionServiceTypeInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
  </operation>
</sequence>
</process>

```

Connect:Enterprise UNIX Server Extract Service

The following table provides an overview of the Connect:Enterprise UNIX Server Extract service:

System name	Connect:Enterprise UNIX Server Extract Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX
Description	This service passes a batch extract request to the Connect:Enterprise UNIX Server adapter. This service works with the adapter to copy data from a Connect:Enterprise UNIX mailbox to the Application. The batch becomes the primary document in the business process.
Business usage	Use this service to take business documents out of a Connect:Enterprise UNIX mailbox and copy them into Application for processing.
Usage example	You want all claims that are added to your mailbox to be translated from EDI into XML and then routed to the appropriate claims processing department. For more information, see <i>How the Connect:Enterprise UNIX Server Extract Service Works</i> on page 269.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Add service, Batch Status service, Log service.
Application requirements	An active instance of Application and Connect:Enterprise UNIX v2.2 or later are required for this service to work.
Initiates business processes?	No
Invocation	Runs as part of a business process.

Business process context considerations	<p>This service can be used in at least two different situations.</p> <ul style="list-style-type: none"> ◆ Where the business process is initiated by a schedule notification from Connect:Enterprise UNIX, also called an auto-connect. In this case, a looping business process is normally used to process each element in the auto-connect definition ACD_INFO/Batches/Batch. (This information is in the process data.) This service can also use XPath statements to find the CEUBatchNumber and CEUBatchDateTime in the process data. Also in this case, the process data automatically includes the CEUServerAdapterInstanceName and a CEUSessionId so that a Connect:Enterprise UNIX Server Begin Session service is not needed. ◆ Where the business process makes an unsolicited request to Connect:Enterprise UNIX, also called a remote connection. In this case this service should be preceded by the Begin Session service. It could also be preceded by a Batch Status service to acquire the necessary CEUBatchNumber and CEUBatchDateTime.
Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified. ◆ Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. ◆ Invalid Session – Indicates an invalid session. ◆ Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. ◆ Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise UNIX Server Adapter. ◆ CEU Service Error – A generic error associated with the service has occurred.
Restrictions	None
Persistence level	All
Testing considerations	For information, see the business process, CEUInterop_DemoPOProcess

How the Connect:Enterprise UNIX Server Extract Service Works

The following describes what happens when this service is used in a business process that is initiated by a Connect:Enterprise UNIX schedule notification.

1. The Connect:Enterprise UNIX Server adapter initiates a business process in response to a schedule notification from Connect:Enterprise UNIX.
2. The business process specifies to extract data from a Connect:Enterprise UNIX mailbox using the Connect:Enterprise UNIX Server Extract service. This service works through the Connect:Enterprise UNIX Server adapter.
3. Using the information from the schedule notification, the Connect:Enterprise UNIX Server adapter extracts the batch that matches the CEUBatchNumber and CEUBatchDateTime in process data.
4. Connect:Enterprise UNIX sets the Transmitted (T) flag on the batch indicating that Application extracted the batch.

5. Application adds a log entry into Connect:Enterprise UNIX that includes the business process ID of this business process.
6. The service adds the batch into the business process as a primary document.
7. Other batch details are added to process data.
For information, see *Output from Service to Business Process* on page 272.
8. Application sets the correlations between the business process that initiated the request and batch summary data from Connect:Enterprise UNIX. These include: CEUBatchId, CEUMailboxId, CEUBatchNumber, schedule definition name and CEUSessionId. These correlations allow subsequent searches for this information using the Central Search function.
9. The system performs additional activities as indicated within the business process. For example, the contents of the batch could be translated into another format.

Implementing the Connect:Enterprise UNIX Server Extract Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Application, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Extract service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server Extract service. For information, see *Configuring the Connect:Enterprise UNIX Server Extract Service* on page 270.
3. Use the Connect:Enterprise UNIX Server Extract service in a business process.

Configuring the Connect:Enterprise UNIX Server Extract Service

To configure the Connect:Enterprise UNIX Server Extract service, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Extract service:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
CEU Server Adapter Instance Name (CEUServerAdapterInstanceName)	<p>The instance of the Connect:Enterprise UNIX adapter that the Extract service should use. The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP.</p>
Document Storage (DatabaseOrFileSystem)	<p>Defines how the document will be stored in the system. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Choose file system when handling very large batches.</p>

GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Extract service in the GPM:

Field	Description
Config	Name of the service configuration. Required.
CEUBatchNumber	Numeric identifier for a batch within Connect:Enterprise UNIX
CEUBatchDateTime	Combined with CEUBatchNumber, this provides unique access to a batch. That is, these two parameters identify what batch is to be extracted.
CEUServerAdapterInstanceName	Normally, this is already defined.
DatabaseOrFileSystem	Normally, this is already defined.

The required parameters CEUBatchNumber and CEUBatchDateTime can be obtained using XPath expressions when the business process is initiated by an auto connect. The following are examples of such XPath expressions:

```
<assign to="CEUBatchNumber"
from="ACD_INFO/Batches/Batch[number(//loop_counter/text())]/CEUBatchNumber/
text()"></assign>

<assign to="CEUBatchDateTime"
from="ACD_INFO/Batches/Batch[number(//loop_counter/text())]/CEUBatchDateTim
e/text()"></assign>
```

Output from Business Process to Service

The following table describes the output from the business process to the Connect:Enterprise UNIX Server Extract service (Message To Service):

Parameter	Description
CEUBatchNumber	Numeric identifier for a batch within Connect:Enterprise UNIX
CEUBatchDateTime	Combined with CEUBatchNumber provides unique access to a batch

Output from Service to Business Process

The following table describes the information that is placed in process data by the Connect:Enterprise UNIX Server Extract service (Message From Service):

Parameter	Description
PrimaryDocument	Batch payload
CEUBatchNumber	Echoes the batch number provided to this service as input.
CEUBatchDateTime	Echoes the batch date and time provided to this service as input.
CEUBatchSize	Size of the message in bytes.
CEUBatchId	Identifier (name) of the batch in Connect:Enterprise UNIX.
CEUMailboxId	The Connect:Enterprise UNIX Mailbox ID from which the batch was extracted.
CEUSessionId	Connect:Enterprise UNIX session that the extract activity is associated with.
CEUProcessFlagStatus	Depicts the flag settings for the batch before the extract occurred. Comma delimited characters. For more information, see your Connect:Enterprise UNIX documentation.

Business Process Example

The following example illustrates a small part of the BPML that utilizes the Connect:Enterprise UNIX Server Extract service:

```
<operation name="CEU Extract Service">
  <participant name="CEUServerExtract"/>
</operation>
```



```
        <output message="ExtractRequest">
            <assign to="CEUBatchNumber"
from="ACD_INFO/Batches/Batch[number(//loop_counter/text())]/CEUBatchNumber/text()"><
/assign>
            <assign to="CEUBatchDateTime"
from="ACD_INFO/Batches/Batch[number(//loop_counter/text())]/CEUBatchDateTime/text()"
></assign>
            <assign to="." from="*" />
<!--            <assign to="DatabaseOrFileSystem">db</assign> -->
        </output>
        <input message="inmsg">
            <assign to="CEUExtractServiceResults" from="*" append="true" />
            <assign to="." from="PrimaryDocument" />
        </input>
    </operation>
```

Connect:Enterprise UNIX Server Log Service

The following table provides an overview of the Connect:Enterprise UNIX Server Log service:

System name	Connect:Enterprise UNIX Server Log Service
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX
Description	This service works with the Connect:Enterprise UNIX Server adapter to write success and error messages to Connect:Enterprise UNIX so they are visible in the Connect:Enterprise UNIX interface for tracking.
Business usage	You use this service in a business process to provide information to Connect:Enterprise UNIX in the event of an error condition that results from the execution of the business process. It can also be used to log success messages.
Usage example	You want the Connect:Enterprise UNIX administrator to be notified when downstream processing of some batch fails or succeeds. For more information, see <i>How the Connect:Enterprise UNIX Server Log Service Works</i> on page 275.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Add service, Extract service, Batch Status service.
Application requirements	An active instance of Application and Connect:Enterprise UNIX v2.2 or later are required for this service to work.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	When the business process is initiated by a schedule notification from Connect:Enterprise UNIX, also called an auto connect, this service should use the CEUAdapterInstanceName and CEUSessionId provided automatically in process data. Normally, a looping business process is used to process each element in the auto connect definition ACD_INFO/Batches/Batch. (This information is in the process data.)

Returned status values	<ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified. ◆ Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. ◆ Invalid Session – Indicates an invalid session. ◆ Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. ◆ Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise UNIX Server Adapter. ◆ CEU Service Error – A generic error associated with the service has occurred.
Restrictions	None
Persistence level	All
Testing considerations	For information, see the business process, CEUInterop_DemoPOProcess

How the Connect:Enterprise UNIX Server Log Service Works

The following describes an example of what happens when this service is used in a business process to log success and error messages to the Connect:Enterprise UNIX interface.

1. A business process extracts a batch from Connect:Enterprise UNIX and passes it along the business process for EDI-to-XML translation.
2. An error occurs during translation and the batch is not translated properly.
3. As a result of the error, the Connect:Enterprise UNIX Server Log service writes a log message about the translation failure to Connect:Enterprise UNIX.
4. When the Connect:Enterprise UNIX administrator views information about the activity, the message provided by the Log service is displayed with a hyperlink to Application.

Implementing the Connect:Enterprise UNIX Server Log Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Application, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Log service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. You can use the pre-installed copy of the Connect:Enterprise UNIX Server Log service, or create a new copy. You might want to create a new copy if you are using more than one copy of the Connect:Enterprise UNIX Server adapter, for example. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Enterprise UNIX Server Log service. For information, see *Configuring the Connect:Enterprise UNIX Server Log Service* on page 276.
4. Use the Connect:Enterprise UNIX Server Log service in a business process.

Configuring the Connect:Enterprise UNIX Server Log Service

To configure the Connect:Enterprise UNIX Server Log service, you must specify field settings in Application and in the GPM. This is only necessary if you create a new copy of the service; the pre-installed copy of the service does not require any configuration.

Application Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Log service:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default)◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. For more information about service groups see <i>Managing Services and Adapters</i> .
CEU Server Adapter Instance Name (CEUServerAdapterInstanceName)	The instance of the Connect:Enterprise UNIX adapter that the Log service should use. The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP.

GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Log service in the GPM:

Field	Description
Config	Name of the service configuration. Required.
CEUBatchId	Name of the batch to which the error or success message relates. Viewed in the Connect:Enterprise UNIX tracking reports as informational. Populates columns in Connect:Enterprise UNIX detailed reporting screens on the administration user interface. Optional
CEUBatchNumber	This is provided to the service as input. Viewed in the Connect:Enterprise UNIX tracking reports as informational. Populates columns in Connect:Enterprise UNIX detailed reporting screens on the administration user interface. Optional

Field	Description
CEUMailboxId	Connect:Enterprise UNIX mailbox where the batch resides. Viewed in the Connect:Enterprise UNIX tracking reports as informational. Populates columns in Connect:Enterprise UNIX detailed reporting screens on the administration user interface. Optional
CEULogStatusCode	The status code that can be viewed in the Connect:Enterprise UNIX tracking statistics column. Success (0) and Error (FA). Required
CEUAdapterInstanceName	Normally, this is already defined.

Output from Business Process to Service

The previous table describes the output from the business process to the Connect:Enterprise UNIX Server Log service (Message To Service).

Business Process Example

The following example illustrates a small part of the BPML that utilizes the Connect:Enterprise UNIX Server Log service.

```
<operation name="CEU Log Service">
  <participant name="CEU Server Log" />
  <output message="LogRequest">
    <assign to="CEUMailboxId"
from="ACD_INFO/Batches/Batch[number(//loop_counter/text())]/CEUMailboxId/text()"></a
ssign>
      <assign to="CEUBatchId"
from="ACD_INFO/Batches/Batch[number(//loop_counter/text())]/CEUBatchId/text()"></ass
ign>
      <assign to="CEUBatchNumber"
from="ACD_INFO/Batches/Batch[number(//loop_counter/text())]/CEUBatchNumber/text()"><
/assign>
      <assign to="CEULogStatusCode">Error</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
```

CPA Lookup Service

The following table provides an overview of the CPA Lookup service:

System name	CPALookup
Graphical Process Modeler (GPM) category	All Services
Description	Retrieves the relevant Collaborative Partner Agreement (CPA) information from an XML file based on an input CPA ID.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Basic status Success – Transformation was successful.◆ Basic status Error – Errors were encountered during transformation or transformation could not be performed. The report contained in the Workflow Context Status report should be consulted for further detail.
Restrictions	The CPA Lookup service can only read information stored in cpa-store.xml.
Testing considerations	<p>To test the CPALookup service, make sure you have set the location of CPA Store correctly. Create a sample business process and execute it.</p> <p>The most frequent problems encountered are:</p> <ul style="list-style-type: none">◆ Invalid CPA Store location◆ Non existent cpa-id, service name, service type and action name in CPA Store◆ Adapter is not active.

Implementing the CPA Lookup Service

To implement the CPA Lookup service, complete the following tasks:

1. Use the pre-installed copy of the CPA Lookup service, or create a new configuration. See *Managing Services and Adapters*.
2. Use the CPA Lookup service in a business process.

Configuring the CPA Lookup Service

There is no configuration necessary for the CPA Lookup service.

Cryptographic Message Service (Build 4315 or higher)

The following table provides an overview of the Cryptographic Message service:

Service name	Cryptographic Message Service
System name	CryptoMsgService
Graphical Process Modeler (GPM) category	All Services
Description	Builds and parses cryptographic messages in SMIME, PEM, or DER format.
Business usage	The Cryptographic Message Service allows users to build and parse cryptographic messages in SMIME, PEM, or DER format.
Usage example	A business process that needs to create or parse the content in a cryptographic message in SMIME, PEM, or DER format can invoke this service by passing the required parameters. Cryptographic messages must follow either Cryptographic Message Syntax or PKCS#7 specification.
Preconfigured?	The Cryptographic Message service should be installed and deployed before it is invoked. However, configuration parameters are not required.
Requires third party files?	Yes. Requires Certicom sbgsepki3.3 jars. This is preloaded in Gentran Integration Suite.
Platform availability	All supported Gentran Integration Suite platforms
Related services	No
Application requirements	No
Initiates business processes?	No. This service does not initiate business process.
Invocation	Yes. Runs as a service within a business process.
Returned status values	<ul style="list-style-type: none">◆ buildResponse - If an exception is thrown during build process, the “exception-message” node is returned to ProcessData with the exception message.◆ parseResponse - If an exception is thrown during parse process, the “exception-message” node is returned to ProcessData with the exception message.
Restrictions	None
Testing considerations	<ul style="list-style-type: none">◆ You should use the right certificates for signing or encryption/decryption.◆ If you receive an error with the condition that certificates used for signing or decrypting are not created with a storepass value of integrator and are created with a keypass value of integrator, see your system administrator.

How the Cryptographic Message Service Works

Cryptographic Message Service (CMS) builds and parses secure messages in Secure MIME (SMIME), Distinguished Encoding Rules (DER), or Privacy Enhanced Email (PEM) format.

The security features of CMS are digital signature and encryption. The Digital signature feature provides authentication, message integrity, and non-denial with proof of origin whereas encryption provides data privacy.

The CMS supports two cryptographic message syntaxes. They are CMS and PKCS#7. If you are building outbound message syntax, you have to indicate the cryptographic message syntax as either one of them. The PKCS#7 uses non-streaming API to handle message building and has limitations to process large files whereas the CMS uses streaming API and has the capability to process large files. If you are parsing an inbound cryptographic message, there is no need to indicate your choice as CMS uses streaming API to parse either PKCS#7 or CMS messages.

Implementing the Cryptographic Message Service

To implement the Cryptographic Message service for use in a business process, complete the following tasks:

1. Create a configuration of the Cryptographic Message service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Cryptographic Message Service*.
2. Specify field settings for the service configuration in the Gentran Integration Suite Admin Console and in the GPM as necessary. For information, see *Configuring the Cryptographic Message Service*.
3. Use the Cryptographic Message service in a business process.

System Administrator Tasks

The following procedures describe the system administrator tasks for cryptographic message service.

Importing a keyCert into Gentran Integration Suite

1. Login to Gentran Integration Suite.
2. Select **Trading Partner** -> **Digital Certificates** -> **System**.
3. Select **Key Certificate** under Check in.
4. Enter the Certificate Name and Private Key Password.
5. Select the certificate and assign an alias to it.
6. Review and click **Finish**. You can use this certificate in your BPML associated with the appropriate field (signingCert or decryptCert).

Importing a Public Certificate into Gentran Integration Suite

1. Login to Gentran Integration Suite.
2. Select **Trading Partner** -> **Digital Certificates** -> **Trusted**.

3. Select **New Certificate** under Check in.
4. Select the certificate and click **Next**.
5. Enter the Certificate Name and **Next**.
6. Review and click **Finish**. You can use this certificate in your BPML associated with the appropriate field (encryptCert or sigVerifyCert).

Configuring the Cryptographic Message Service

You can create one service instance for building and parsing cryptographic messages. You can configure the service in Gentran Integration Suite and also in the GPM.

To configure the Cryptographic Message service, you must specify settings for the following fields:

Note: Any field values passed from a prior service can override any of configured fields for this service.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None - You do not want to include this configuration in a group at this time (default) ◆ Create New Group - You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group - If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>
Cryptographic Message Syntax	<p>Drop-down menu containing a list of cryptographic message syntaxes for building cryptographic messages. Required.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ CMS (default) ◆ PKCS#7

Field	Description
Security Type	<p>Drop-down menu containing the security type for building cryptographic messages. Required.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ Encrypted Only (default) - Encrypts the message only ◆ Detached Signed Only - Signs the original document and leaves the signature detached from the original document. If the message output format is SMIME, multipart MIME message will separate the original document and signature. If the message output format is DER or PEM, only detached signature will be returned by the service. ◆ Embedded Signed Only - Signs the original document and embeds the original document inside the signature. ◆ Detached Signed and Encrypted - Creates detached signed signature and encrypts the signed message. If the message output format is SMIME, the encryption is applied on the multipart MIME message. If the message output format is DER or PEM, the encryption is applied on the detached signature only. ◆ Embedded Signed and Encrypted - Creates embedded signed signature and encrypts the signed message.
Message Output Format	<p>Message output format for generating the signed or encrypted message. Required.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ SMIME (default) - The signed or encrypted message will be output in MIME format. ◆ DER - The signed or encrypted message will be output in DER encoded format. ◆ PEM - The signed or encrypted message will be output in PEM encoded format, which is a base64 encoded DER format and enclosed between a start and an end boundary.
Document MIME Content Type	<p>This parameter is enabled only if you select SMIME as the message output format.</p> <p>MIME content type for the document that needs to be packaged. If the input document is set with the content type, the value will override the setting here. Optional.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ application (default) ◆ text ◆ message ◆ image ◆ video ◆ audio

Field	Description
Document MIME Sub Content Type	<p>This parameter is enabled only if you select SMIME as the message output format</p> <p>MIME sub content type for the document that needs to be packages. If the input document is set with the sub content type, the value will override the setting here. Optional.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ octet-stream (default) ◆ plain ◆ edi-x12 ◆ edifact ◆ edi-consent ◆ xml
Content Transfer Encoding	<p>This parameter is enabled only if you select SMIME as the message output format.</p> <p>Content transfer encoding format. Optional.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ Base64 (default) ◆ None
Apply Content Transfer Encoding on Detached Document	<p>This parameter is enabled only if you select SMIME as the message output format.</p> <p>To indicate if content transfer encoding should be applied on the detached document. This is used for <i>Detached Signed Only</i> and <i>Detached Signed and Encrypted</i> security types. Optional.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ Yes (default) ◆ No
Encryption Algorithm:	<p>Content encryption algorithm. Optional.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ Triple DES (3DES) 168 CBC with PKCS5 padding (default) ◆ 56-bit DES CBC with PKCS5 padding ◆ 128-bit RC2 CBC with PKCS5 padding ◆ 40-bit RC2 CBC with PKCS5 padding ◆ 128-bit AES CBC with PKCS5 padding ◆ 192-bit AES CBC with PKCS5 padding ◆ 256-bit AES CBC with PKCS5 padding
Encryption Certificate(s):	<p>Public certificates to encrypt the document. A list or a single certificate can be chosen to encrypt the same document. When you choose multiple certificates, it allows multiple recipients to decrypt the message. Optional.</p>

Field	Description
Signature Options	Options to sign the message. Required. Valid values are: <ul style="list-style-type: none"> ◆ Single Signature (default) ◆ Multiple Signatures ◆ Counter Signature ◆ No Signature Required
Signing Algorithm	The signing algorithm to hash the document. Optional. Valid values are: <ul style="list-style-type: none"> ◆ SHA1 (default) ◆ MD5
Signing Certificate(s)	Private certificates to sign the document. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Select a signing certificate if you have selected Single Signature. ◆ Select a list of signing certificates for multiple users to sign the document if you have selected Multiple Signatures. ◆ Select a list of signing certificates for multiple users to sign the document and countersign the signature if you have selected Counter Signature.
Message Input Format	Message input format for parsing the signed or encrypted message. Required. Valid values are: <ul style="list-style-type: none"> ◆ SMIME (default) ◆ DER ◆ PEM
Security Type	This parameter is enabled only if you select either PEM or DER as the message input format. Security type that is applied to the inbound cryptographic message. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Encrypted Only (default) - The inbound message is encrypted only. ◆ Detached Signed Only - The inbound message is signed in detached format. ◆ Embedded Signed Only - The inbound message is signed in embedded format. ◆ Detached Signed and Encrypted - The inbound message is signed in detached format and then encrypted. ◆ Embedded Signed and Encrypted - The inbound message is signed in embedded format and then encrypted.
Decryption Certificate	Private certificate to decrypt the cryptographic message. Optional.

Field	Description
Signature Verification Certificate(s)	Public certificates to verify signed cryptographic message. Optional. Note: You can select single certificate if the inbound message is signed by one certificate or select a list of certificates if multiple certificates sign the inbound message. Based on the certificates list sequence, counter signature verification starts from the first level of the signature.

Parameters That Must Be Added in BPML

The following additional parameters are available for use with the Cryptographic Message service, but can only be added by editing your business process manually. This parameter is not available through the Admin console or the GPM:

Parameter	Description
Action	The two values are either <i>build</i> or <i>parse</i> . Required.
pipelineTimeout	Controls the duration of building or parsing process. By default, the value is 300 seconds and can be increased to process large files. Optional.

Business Process Examples

The parameters passed from the BPML precede over the parameters passed from the service. The following BPML examples illustrate using the cryptographic message service instance:

Example Business Process 1

The following BPML builds the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```
<process name="cryptomsg_build">
  <sequence>
    <operation name="Crypto Message Service">
      <participant name="CryptoMsgService"/>
      <output message="buildRequest">
        <assign to="." from="*" />
        <assign to="action">build</assign>
      </output>
      <input message="buildResponse">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Example Business Process 2

The following BPML parses the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```
<process name="cryptomsg_parse">
  <sequence>
    <operation name="Crypto Message Service">
      <participant name="CryptoMsgService"/>
    </operation>
  </sequence>
</process>
```

```

    <output message="parseRequest">
      <assign to="." from="*" />
      <assign to="action">parse</assign>
    </output>
    <input message="parseResponse">
      <assign to="." from="*" />
    </input>
  </operation>
</sequence>
</process>

```

Example Business Process 3

The following BPML builds and parses the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```

<process name="cryptomsg_buildandparse">
  <sequence>
    <operation name="Crypto Message Service">
      <participant name="CryptoMsgService" />
      <output message="buildRequest">
        <assign to="." from="*" />
        <assign to="action">build</assign>
        <!-- securityType=3 Encrypted Only,
            securityType=1 Detached Signed Only,
            securityType=2 Embedded Signed Only,
            securityType=4 Detached Signed and Encrypted,
            securityType=5 Embedded Signed and Encrypted -->
        <assign to="securityType">4</assign>
        <!-- signOptions=0 No Signature Required,
            signOptions=1 Single Signature,
            signOptions=2 Multiple Signatures,
            signOptions=3 Counter Signature -->
        <assign to="signOptions">3</assign>
        <assign to="signAlgo">SHA1</assign>
        <assign to="signCerts">smime_priv1,smime_priv2,smime_priv3</assign>
        <!-- encryption algorithm
            encAlgo=0 Triple DES 168 CBC with PKCS5 padding
            encAlgo=1 56-bit DES CBCwith PKCS5 padding
            encAlgo=2 128-bit RC2 CBC with PKCS5 padding
            encAlgo=4 40-bit RC2 CBC with PKCS5 padding
            encAlgo=6 128-bit AES CBC with PKCS5 padding
            encAlgo=7 192-bit AES CBC with PKCS5 padding
            encAlgo=8 256-bit AES CBC with PKCS5 padding -->
        <assign to="encAlgo">0</assign>
        <assign to="encCerts">smime_pub1,smime_pub2</assign>
      </output>
      <input message="buildResponse">
        <assign to="." from="*" />
      </input>
    </operation>

    <operation name="Crypto Message Service">
      <participant name="CryptoMsgService" />
      <output message="parseRequest">
        <assign to="." from="*" />
        <assign to="action">parse</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

        <assign to="verifyCerts">smime_pub3,smime_pub2,smime_pub1</assign>
<assign to="decryptCert">smime_priv1</assign>
    </output>
    <input message="parseResponse">
        <assign to="." from="*" />
    </input>
</operation>
</sequence>
</process>

```

Example Business Process 4

The following BPML puts the detached document under the *detachedDoc* area when parsing detaching only inbound message in PEM or DER format.

```

<process name="cryptomsg_parse">
  <sequence>
    <operation name="Import Document Request">
      <participant name="CryptoMsgTestFSA" />
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder" from="'/gisinstall'"/>
        <assign to="filter" from="'detached_doc.txt'"/>
        <assign to="useSubFolders">false</assign>
        <assign to="bootstrap">false</assign>
        <assign to="deleteAfterCollect">false</assign>
        <assign to="." from="*" />
      </output>
      <input message="FileSystemOutputMessage">
        <assign to="." from="*" />
      </input>
    </operation>
    <assign to="detachedDoc" from="PrimaryDocument/@SCIOBJECTID"/>
    <operation name="Import Document Request">
      <participant name="CryptoMsgTestFSA" />
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder" from="'/gisinstall'"/>
        <assign to="filter" from="'signed_msg.txt'"/>
      <assign to="useSubFolders">false</assign>
      <assign to="bootstrap">false</assign>
      <assign to="deleteAfterCollect">false</assign>
      <assign to="." from="*" />
    </output>
    <input message="FileSystemOutputMessage">
      <assign to="." from="*" />
    </input>
  </operation>
  <operation name="Crypto Message Service">
    <participant name="CryptoMsgService" />
    <output message="parseRequest">
      <assign to="." from="*" />
      <assign to="action">parse</assign>
      <!--securityType=3 Encrypted Only,
securityType=1 Detached Signed Only,
securityType=2 Embedded Signed Only,
securityType=4 Detached Signed and Encrypted,

```


Output from Service to Business Process

The following table describes the output from the cryptographic message service to the BPML ProcessData, when the service action is “build”:

Scenario	Output
Certificates used for encryption are acceptable	

Scenario	Output
-----------------	---------------

Certificate used for encryption or signing has expired

Or

Certificate used for encryption has been revoked

Certificate used for encryption fails to process. For example, if the encryption certificate is not found in Gentran Integration Suite.

The following table describes the output from the cryptographic message service to BPML ProcessData, when the service action is "parse":

Scenario	Output
-----------------	---------------

Decryption is passed

Scenario**Output**

Decryption certificate
not found in Gentran
Integration Suite

Decryption certificate
failed to decrypt

Signature verification
is passed

Signature verification
fails

Scenario	Output
----------	--------

Multiple signature verification fails	
---------------------------------------	--

Signature verification certificate is revoked	
---	--

The CMS service allows you to use an expired certificate to encrypt/decrypt or sign/verify the message if “validity” flag is not enabled when you check in the certificate into the system. The certificate status and expiry time is shown in the ProcessData as part of CMS service output.

The certificate ExpiryTime and SigningTime is displayed in UTC timezone in yyyyMMddHHmmssZ format. The BPML can perform the following checks after calling the CMS service:

- ExpiryTime against SigningTime to determine if the signature verified by the expired certificate is acceptable or not.

- ExpiryTime against the current date to determine if the encrypted or signed data created the expired certificate is acceptable or not.

Data Sweeper Service (Build 4319 or higher)

The following table provides an overview of the Data Sweeper service:

System name	Data Sweeper Service
Graphical Process Modeler (GPM) categories	System Services
Description	The Data Sweeper service is an optionally scheduled system service that cleans up data that is not in use and not cleaned by other system clean up processes due to lack of any continued associations to the data.
Business usage	The Data Sweeper service is a system service that corrects discovered entity relationship issues within the database that could potentially cause performance and unnecessary database expansion.
Usage examples	Based on the optional parameters and file settings, you have set, Data Sweeper service cleans up the data potentially left from data disassociations from the following tables: <ul style="list-style-type: none">◆ EDIINT◆ Correlations◆ Document Clones◆ Document Life Span◆ GUID◆ Performance Engine Stats◆ Workflow Context◆ Workflow ID Note: The Data Sweeper command line option (or) also cleans DataTableConfig on the recommendation of the Sterling Customer Support.
Preconfigured?	Yes. DataSweeper.
Requires third party files?	No

Platform availability	The following platforms are supported: <ul style="list-style-type: none"> ◆ HP-UX ◆ IBM AIX ◆ IBM iSeries (OS/400) ◆ Microsoft Windows 2000 ◆ RedHat AS ◆ Sun Solaris ◆ United Linux
Related services	None Note: The Data Sweeper service references the <code>noapp.properties</code> file in the <code><GISInstallDir>.bin.properties</code> file.
Application requirements	None
Initiates business processes?	Data Sweeper service is a system service that runs a business process. You can run <code>DataSweeper.sh</code> or <code>DataSweeper.cmd</code> from the command line.
Invocation	The Data Sweeper service is not for use in customer business processes but you may use it in a system business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ Success ◆ Failure
Restrictions	To run the <code>DataTableConfig</code> command line option, ensure that the Application is shut down, and the database is running. Use the <code>DataTableConfig</code> option only if Customer Support advises you to run it. Note: Data Sweeper service starts MySQL if it is already not running.
Persistence level	The default persistence level is Full. You can set the persistence to a lower level if logging is not required.
Testing considerations	Data Sweeper service writes to the <code>noapp.properties</code> log file.

Configuring the Data Sweeper Service

There are no configurable parameters. All options must be set in the BPML, at the command line, or in the properties file. At run time, the command line or BPML will override the properties file settings in the case of a conflict.

Business Process Example

The following example business process illustrates using the Data Sweeper service:

```

<rule name="obtainLock">
  <condition>controlLock ='true' </condition>
</rule>
<sequence>
<assign to='controlLock'>>false</assign>

<operation name="SetLock">
  <participant name="SystemLockService"/>
  <output message="Xout">
    <assign to="LOCK_KEY">DataSweeper</assign>
    <assign to="DURATION">86400000</assign>
<assign to="CLEAR_ON_START_UP">>true</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<assign to='controlLock'>>true</assign>

<operation name="Service">
  <participant name="DataSweeper"/>
  <output message="Xout">
    <assign to="batchSize">5000</assign>
    <assign to="autocorrect">TRUE</assign>
    <assign to="maxIterations">1000</assign>
    <assign to="sweeperTimeout">1080000</assign>
    <assign to="sweeperTimeoutThreshold">36000000</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="unLock">
  <participant name="SystemLockService"/>
  <output message="Xout">
    <assign to="ACTION">unlock</assign>
    <assign to="LOCK_KEY">DataSweeper</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<onFault>
  <sequence name="LockFaield">
    <choice>
      <select>
        <case ref="obtainLock" activity="proceedWithLocking"/>
        <case ref="obtainLock" negative="true" activity="stopWithoutLocking"/>
      </select>
    </choice>
    <sequence name="proceedWithLocking">

```

```

    <operation>
      <participant name="SystemLockService"/>
      <output message="Xout">
        <assign to="ACTION">unlock</assign>
        <assign to="LOCK_KEY">DataSweeper</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
  <sequence name="stopWithoutLocking">
    <assign to="Document/Msg" append="true">Failed to obtain a lock.!!</assign>
  </sequence>
</choice>
<assign to="Document/Status" append="true">Failed!</assign>
<assign to="Document/Msg" append="true">DataSweeper failed!</assign>
<assign to="Document/CurrentHost" append="true">loki</assign>
<assign to="Document/CurrentPort" append="true">53000</assign>
<assign to="Document/DetailMsg" from="/ProcessData/StatusRpt/text()"
append="true"></assign>

  <operation name="SMTP Send">
    <participant name="SMTP_SEND_ADAPTER"/>
    <output message="SMTP_SEND_ADAPTERInputMessage">
      <assign
to="xport-smtp-mailfrom">alert_email_recipient@yournet.com</assign>
      <assign to="xport-smtp-mailhost">yourmailhost.local</assign>
      <assign to="xport-smtp-mailport">25</assign>
      <assign to="xport-smtp-mailto">alert_email_recipient@yournet.com</assign>
      <assign to="xport-smtp-mailsubject">Automated Event Notification -
DataSweeper Failed</assign>
      <assign to="PrimaryDocument" from="DOMToDoc(Document)/@*"></assign>
      <assign to="." from="*"></assign>
    </output>

    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
</onFault>

</sequence>
</process>

```

DB Monitor Service (Build 4300 - Build 4321)

The DB Monitor service monitors the database for common conditions that may be signs of problems with database health. The following table provides an overview of the DB Monitor service:

System Name	DBMonitorService
Graphical Process Modeler (GPM) categories	All Services
Description	This service monitors the database for common conditions that may be signs of system problems. May be scheduled to run at regular intervals.
Business usage	The DB Monitor service is used by the Schedule_DBMonitorService business process. Together they are used by the system to notify the system administrator if key database health thresholds are exceeded or if maintenance is needed. This maintenance includes indexes, rebuilds, and analyze table actions. By default, this service is scheduled to run every Monday at 4:00 A.M.
Usage example	The DB Monitor service is scheduled to run once a week. If the service determines that a key indicator has exceeded its threshold, the system will notify the system administrator by e-mail. It also monitors the database maintenance needs and provides the instructions (SQL commands) necessary to rebuild indexes and analyze tables.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	The database must support the requested functionality. Functionality available is dependent on the database's ability to support it. For example, DB2 and MySQL do not support providing statistics on database size.
Initiates business processes?	The DB Monitor service invokes an associated system business process, Schedule_DBMonitorService.
Invocation	Scheduler
Business process context considerations	None
Returned status values	N/A
Restrictions	None
Persistence level	System Default
Testing considerations	None

How the DB Monitor Service Works

The DB Monitor service sends notification, as configured by the Application event alert system, of any recommended maintenance. If the DB Monitor service determines that the system administrator needs to rebuild the index, or analyze database tables, the service will produce a primary document that provides information about the recommended maintenance. The DB Monitor service will also provide notification if key database health thresholds are exceeded. By default, the system will send an e-mail to the system administrator.

The preconfigured instance of the DB Monitor service, named `DBMonitorService`, has an associated business process named `Schedule_DBMonitorService` that performs all the available actions of the DB Monitor service using default values for each. The `DBMonitorService` instance is scheduled to run each Monday at 4:00 A.M. You can change the run day and time using the Scheduler.

If you create a new instance of the DB Monitor service and specify a schedule in the service configuration, the system will create an associated business process using the name of your instance (minus any spaces) preceded by `Schedule_`. For example, if you name your instance *My DB Monitor Service*, the business process name will be `Schedule_MyDBMonitorService`. If you do not use a schedule in the service configuration, a business process will not automatically be created.

You can change settings by editing the BPML in the associated business process. You can also use the DB Monitor service in your own business process and specify settings in the Graphical Process Monitor, or directly in your BPML.

Implementing the DB Monitor Service

To implement the DB Monitor service, complete the following tasks:


1. Create a new configuration of the DB Monitor service. For basic information about creating service configurations, see *Managing Services and Adapters*.
2. Specify field settings for scheduling the service configuration in the Application Admin Console.
3. If necessary, specify settings in the GPM, or using BPML in your business processes. This step is not necessary if you want to use the pre-configured business process with default settings.

Configuring the DB Monitor Service

Creating a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the DB Monitor service.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about service groups, see <i>Managing Services and Adapters</i>.</p>
Run As User	<p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid values: Any valid Application user ID.</p>
Use 24 Hour Clock Display	<p>Check the box to use the 24-hour clock instead of the default 12-hour clock.</p>
Schedule	<p>Specify the scheduling information for running the Auto Terminate service.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If you select this field, the service does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service each day. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. <p>Note: Any value except Do not use schedule will cause a <code>Schedule_InstanceName</code> business process to be automatically created. See <i>How the DB Monitor Service Works</i> on page 298 for more information. If you select Do not use schedule, you must specify service settings in the GPM.</p>

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
ACTION	<p>Specifies an action to take. Required. Valid Values:</p> <ul style="list-style-type: none">◆ Check DB Full – Check the current size of the database and provide the result as a percentage of the maximum database size.◆ Check DB Grown – Check how much the database has grown since the last time it was checked and provide the result as a percentage.◆ Check DB Index – Check whether indexes need to be rebuilt or check whether tables need to be analyzed. Either REBUILD_INDEXES or ANALYZE must be set to true. <p>Note: The Schedule_ <i>instanceName</i> business process that is automatically created with an implementation of the DB Monitor service performs all three actions using the default values for each.</p>
ANALYZE	<p>Used only if ACTION is set to Check DB Index. Specifies whether to check whether tables need to be analyzed. Optional. Valid values:</p> <ul style="list-style-type: none">◆ true – Check whether tables need to be analyzed.◆ false – Do not check whether tables need to be analyzed. Default.
DO_All_TABLE	<p>Used only if ACTION is set to Check DB Index. Specifies whether to check all the tables from the database. Optional. Valid values:</p> <ul style="list-style-type: none">◆ true – Check all the database tables.◆ false – Check the tables that recorded on the database since last time rebuilds Default.
LAST_ANALYZED_DATE	<p>Used only if ACTION is set to Check DB Index. The expiration date and time to check against the database. If the table is expired since the last check, it will placed in a list for index rebuilding or analysis. Optional. Example value: 20051230-12:12:12.000. Default is current date.</p>
MAX_PERCENT_FULL	<p>Used only if ACTION is set to Check DB Full. The percentage of maximum database size the database must reach before a notification is sent to the event framework. Optional. Default is 80.</p>
MAX_PERCENT_GROWN	<p>Used only if ACTION is set to Check DB Grown. The percentage of maximum database size that the database must have grown since last time checked before a notification is sent to the event framework. Optional. Default is 30.</p>
REBUILD_INDEXES	<p>Used only if ACTION is set to Check DB Index. Specifies whether to check whether the indexes need rebuilt. Optional. Valid values:</p> <ul style="list-style-type: none">◆ true – Check whether indexes need to be rebuilt.◆ false – Do not check whether indexes need to be rebuilt. Default.

Field	Description
SAVE_STATUS	<p>Used only if ACTION is set to For Check DB Full or Check DB Grown. Specifies whether to save the database usage to the database to allow comparison of the difference for the next check. Optional. Valid values</p> <ul style="list-style-type: none"> ◆ true – Save the database usage to the database. ◆ false – Do not save the database usage to the database. Default.

Business Process Example

The following example illustrates a business process containing the DB Monitor service. The following BPML is from the Schedule_DBMonitorService business process invoked by the DB Monitor service.

```
<process name="Schedule_DBMonitorService">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="SystemLockService"/>
      <output message="Xout">
        <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="CLEAR_ON_START_UP">>true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Check for database percentage grown beyond threshold since last check

```
<sequence name="Process">
  <operation name="DBMonitorService_1">
    <participant name="DBMonitorService"/>
    <output message="Xout">
      <assign to="ACTION">CHECK_DB_GROWN</assign>
      <assign to="MAX_PERCENT_GROWN">30</assign>
      <assign to="SAVE_STATUS">>true</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
```

Check for database percentage full.

```
<operation name="DBMonitorService_2">
  <participant name="DBMonitorService"/>
  <output message="Xout">
    <assign to="ACTION">CHECK_DB_FULL</assign>
    <assign to="MAX_PERCENT_FULL">80</assign>
    <assign to="SAVE_STATUS">>true</assign>
    <assign to="." from="*"></assign>
  </output>
</operation>
```

```

    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

```

Check for whether or not the database should be re-indexed

```

<operation name="DBMonitorService_3">
  <participant name="DBMonitorService"/>
  <output message="Xout">
    <assign to="ACTION">CHECK_DB_INDEX</assign>
    <assign to="REBUILD_INDEXES">true</assign>
    <assign to="ANALYZE">true</assign>
    <assign to="DO_All_TABLE">>false</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<sequence name="UnLock">
  <operation name="UnLock">
    <participant name="SystemLockService"/>
    <output message="Xout">
      <assign to="ACTION">unlock</assign>
      <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <onFault>
    <assign to="UnLock_Msg" append="true">Failed to obtain a unlock!</assign>
  </onFault>
</sequence>
<onFault>
  <operation>
    <participant name="SystemLockService"/>
    <output message="Xout">
      <assign to="ACTION">unlock</assign>
      <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</onFault>
</sequence>
<onFault>
  <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
</onFault>

```

```
</sequence>  
</process>
```

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the DB Monitor service to the business process:

Parameter	Description
UsagePercentFull	The current percent used of maximum database size.
UsageUsed	The current amount of space, in MB, used by the database.
UsageTotal	The total amount of space, in MB, used by the database.
UsageUsedKB	The current amount of space, in KB, used by the database.
PercentGrown	The percentage the database has grown since the last time it was checked.
SQL_<Number_of_SQL>	The actual SQL statements for index rebuilds and analyze tables.

DB Monitor Service (Build 4322 or higher)

The DB Monitor service monitors the database for common conditions that may be signs of problems with database health. The following table provides an overview of the DB Monitor service:

System Name	DBMonitorService
Graphical Process Modeler (GPM) categories)	All Services
Description	This service monitors the database for common conditions that may be signs of system problems. May be scheduled to run at regular intervals.
Business usage	The DB Monitor service is used by the Schedule_DBMonitorService business process. Together they are used by the system to notify the system administrator if key database health thresholds are exceeded or if maintenance is needed. This maintenance includes indexes, rebuilds, and analyze table actions. By default, this service is scheduled to run every Monday at 4:00 A.M.
Usage example	The DB Monitor service is scheduled to run once a week. If the service determines that a key indicator has exceeded its threshold, the system will notify the system administrator by e-mail. It also monitors the database maintenance needs and provides the instructions (SQL commands) necessary to rebuild indexes and analyze tables.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	The database must support the requested functionality. Functionality available is dependent on the database's ability to support it. For example, DB2 and MySQL do not support providing statistics on database size.
Initiates business processes?	The DB Monitor service invokes an associated system business process, Schedule_DBMonitorService.
Invocation	Scheduler
Business process context considerations	None
Returned status values	N/A
Restrictions	None
Persistence level	System Default
Testing considerations	None

How the DB Monitor Service Works

The DB Monitor service sends notification, as configured by the Application event alert system, of any recommended maintenance. If the DB Monitor service determines that the system administrator needs to rebuild the index, or analyze database tables, the service will produce a primary document that provides information about the recommended maintenance. The DB Monitor service will also provide notification if key database health thresholds are exceeded. By default, the system will send an e-mail to the system administrator.

The preconfigured instance of the DB Monitor service, named `DBMonitorService`, has an associated business process named `Schedule_DBMonitorService` that performs all the available actions of the DB Monitor service using default values for each. The `DBMonitorService` instance is scheduled to run each Monday at 4:00 A.M. You can change the run day and time using the Scheduler.

If you create a new instance of the DB Monitor service and specify a schedule in the service configuration, the system will create an associated business process using the name of your instance (minus any spaces) preceded by `Schedule_`. For example, if you name your instance *My DB Monitor Service*, the business process name will be `Schedule_MyDBMonitorService`. If you do not use a schedule in the service configuration, a business process will not automatically be created.

You can change settings by editing the BPML in the associated business process. You can also use the DB Monitor service in your own business process and specify settings in the Graphical Process Monitor, or directly in your BPML.

Implementing the DB Monitor Service

To implement the DB Monitor service, complete the following tasks:


1. Create a new configuration of the DB Monitor service. For basic information about creating service configurations, see *Managing Services and Adapters*.
2. Specify field settings for scheduling the service configuration in the Application Admin Console.
3. If necessary, specify settings in the GPM, or using BPML in your business processes. This step is not necessary if you want to use the pre-configured business process with default settings.

Configuring the DB Monitor Service

Creating a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the DB Monitor service.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about service groups, see <i>Managing Services and Adapters</i>.</p>
Run As User	<p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid values: Any valid Application user ID.</p>
Use 24 Hour Clock Display	<p>Check the box to use the 24-hour clock instead of the default 12-hour clock.</p>
Schedule	<p>Specify the scheduling information for running the Auto Terminate service.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If you select this field, the service does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service each day. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. <p>Note: Any value except Do not use schedule will cause a <i>Schedule_InstanceName</i> business process to be automatically created. See <i>How the DB Monitor Service Works</i> on page 305 for more information. If you select Do not use schedule, you must specify service settings in the GPM.</p>

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
ACTION	<p>Specifies an action to take. Required. Valid Values:</p> <ul style="list-style-type: none">◆ Check DB Full – Check the current size of the database and provide the result as a percentage of the maximum database size.◆ Check DB Grown – Check how much the database has grown since the last time it was checked and provide the result as a percentage.◆ Check DB Index – Check whether indexes need to be rebuilt or check whether tables need to be analyzed. Either REBUILD_INDEXES or ANALYZE must be set to true. <p>Note: The Schedule_ <i>instanceName</i> business process that is automatically created with an implementation of the DB Monitor service performs all three actions using the default values for each.</p>
ANALYZE	<p>Used only if ACTION is set to Check DB Index. Specifies whether to check whether tables need to be analyzed. Optional. Valid values:</p> <ul style="list-style-type: none">◆ true – Check whether tables need to be analyzed. Default.◆ false – Do not check whether tables need to be analyzed.
EXECUTE_QUERY	<p>Used only if ACTION is set to Check DB Index. Specifies whether table indexes will be rebuilt. Optional. Valid values:</p> <p>true - Rebuild table indexes automatically.</p> <p>false - Do not rebuild table indexes. Output queries to run manually. Default.</p>
DO_AII_TABLE	<p>Used only if ACTION is set to Check DB Index. Specifies whether to check all the tables from the database. Optional. Valid values:</p> <ul style="list-style-type: none">◆ true – Check all the database tables.◆ false – Check the tables that recorded on the database since last time rebuilds Default.
LAST_ANALYZED_DATE	<p>Used only if ACTION is set to Check DB Index. The expiration date and time to check against the database. If the table is expired since the last check, it will placed in a list for index rebuilding or analysis. Optional. Example value: 20051230-12:12:12.000. Default is current date.</p>
MAX_PERCENT_FULL	<p>Used only if ACTION is set to Check DB Full. The percentage of maximum database size the database must reach before a notification is sent to the event framework. Optional. Default is 80.</p>
MAX_PERCENT_GROWN	<p>Used only if ACTION is set to Check DB Grown. The percentage of maximum database size that the database must have grown since last time checked before a notification is sent to the event framework. Optional. Default is 30.</p>

Field	Description
REBUILD_INDEXES	Used only if ACTION is set to Check DB Index . Specifies whether to check whether the indexes need rebuilt. Optional. Valid values: <ul style="list-style-type: none"> ◆ true – Check whether indexes need to be rebuilt. ◆ false – Do not check whether indexes need to be rebuilt. Default.
SAVE_STATUS	Used only if ACTION is set to For Check DB Full or Check DB Grown . Specifies whether to save the database usage to the database to allow comparison of the difference for the next check. Optional. Valid values <ul style="list-style-type: none"> ◆ true – Save the database usage to the database. ◆ false – Do not save the database usage to the database. Default.

Business Process Example

The following example illustrates a business process containing the DB Monitor service. The following BPML is from the Schedule_DBMonitorService business process invoked by the DB Monitor service.

```
<process name="Schedule_DBMonitorService">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="SystemLockService"/>
      <output message="Xout">
        <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="CLEAR_ON_START_UP">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Check for database percentage grown beyond threshold since last check

```
<sequence name="Process">
  <operation name="DBMonitorService_1">
    <participant name="DBMonitorService"/>
    <output message="Xout">
      <assign to="ACTION">CHECK_DB_GROWN</assign>
      <assign to="MAX_PERCENT_GROWN">30</assign>
      <assign to="SAVE_STATUS">true</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
```

Check for database percentage full.

```
<operation name="DBMonitorService_2">
  <participant name="DBMonitorService"/>
</operation>
```

```

<output message="Xout">
  <assign to="ACTION">CHECK_DB_FULL</assign>
  <assign to="MAX_PERCENT_FULL">80</assign>
  <assign to="SAVE_STATUS">true</assign>
  <assign to="." from="*"></assign>
</output>
<input message="Xin">
  <assign to="." from="*"></assign>
</input>
</operation>

```

Check for whether or not the database should be re-indexed

```

<operation name="DBMonitorService_3">
  <participant name="DBMonitorService"/>
  <output message="Xout">
    <assign to="ACTION">CHECK_DB_INDEX</assign>
    <assign to="REBUILD_INDEXES">true</assign>
    <assign to="ANALYZE">true</assign>
    <assign to="EXECUTE_QUERY">false</assign>
    <assign to="DO_All_TABLE">false</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<sequence name="UnLock">
  <operation name="UnLock">
    <participant name="SystemLockService"/>
    <output message="Xout">
      <assign to="ACTION">unlock</assign>
      <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <onFault>
    <assign to="UnLock_Msg" append="true">Failed to obtain a unlock!</assign>
  </onFault>
</sequence>
<onFault>
  <operation>
    <participant name="SystemLockService"/>
    <output message="Xout">
      <assign to="ACTION">unlock</assign>
      <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

```

```

        </operation>

    </onFault>
</sequence>
<onFault>
    <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
</onFault>
</sequence>
</process>

```

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the DB Monitor service to the business process:

Parameter	Description
UsagePercentFull	The current percent used of maximum database size.
UsageUsed	The current amount of space, in MB, used by the database.
UsageTotal	The total amount of space, in MB, used by the database.
UsageUsedKB	The current amount of space, in KB, used by the database.
PercentGrown	The percentage the database has grown since the last time it was checked.
SQL_<Number_of_SQL>	The actual SQL statements for index rebuilds and analyze tables.

Document Keyword Replace Service

The Document Keyword Replace service enables you to find and replace text within a document.

The following table provides an overview of the Document Keyword Replace service:

System name	Document Keyword Replace Service
Graphical Process Modeler (GPM) category	All Services
Description	The Document Keyword Replace Service allows you to replace user-defined keywords in a document. This can be used to manipulate hexadecimal or string values in a document. A keyword can be replaced with either a replacement string or the contents of a document.
Preconfigured?	A preconfigured instance is available for use with the SyncEngine.
Business usage	Because of disparities between how different trading partners' enterprise systems process data, the format of the data that is exchanged between trading partners often needs to be manipulated before sending or after receiving or in some cases, on both sides of the transmission.
Usage example	Assume that the VAN used by your trading partner expects each line of data to be terminated with a Carriage Return and Line Feed, and the data from your enterprise system is terminated by tilde (~) characters. You would use the DocKeywordReplace service to convert the tildes into Carriage Return / Line Feed characters, before transmitting the data to your trading partner.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	This service uses the primary document.
Returned status values	<ul style="list-style-type: none">◆ Success – Document Keyword Replace service was successful.◆ Error – Document Keyword Replace service was unsuccessful.
Restrictions	None

Implementing the Document Keyword Replace Service

To implement the Document Keyword Replace service, complete the following tasks:

1. Create a Document Keyword Replace service configuration. See *Creating a Service Configuration*.
2. Configure the Document Keyword Replace service. See *Configuring the Document Keyword Replace Service* on page 312.
3. Use the Document Keyword Replace service in a business process.

Configuring the Document Keyword Replace Service

To configure the Document Keyword Replace service, use the following BPML parameters in your business process:

BPML	Description
keyword<x>	<p>Keyword that the service should look for. Required.</p> <p><x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: keyword1, keyword2, keyword3, etc.</p> <p>Note: Each keyword must have a corresponding replace. Otherwise, it will be ignored and you may receive an error message. All subsequent keyword<x>/replace<x> pairs will also be ignored.</p> <p>Note: Always define each keyword<x>/replace<x> pair sequentially. For example, do not specify a keyword2/replace2 pair and a keyword4/replace4 pair without a keyword3/replace3 pair between them.</p>
keywordtype<x>	<p>Type of keyword. Optional. Valid values are hex, string, and encoding type for the parameter keyword<x>. Encoding type is any supported encoding format such as UTF-8 and UTF-16. Default is string.</p> <p>Note: Encoding is supported only when literal_mode is set to TRUE.</p> <p>The following example demonstrates the use of UTF-8 as the encoding type for the parameter keyword<x>:</p> <pre><output message="DocKeywordReplaceInputMessage"> <assign to="literal_mode">true</assign> <assign to="keyword1" from="'&#x6E;'></assign> <assign to="keywordtype1">UTF-8</assign> <assign to="replace1" from="'&#x7F;'></assign> <assign to="replacetype1">UTF-8</assign> </output></pre> <p><x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: keywordtype1, keywordtype2, keywordtype3, etc.</p> <p>Note: This parameter is optional in either mode.</p> <p>Note: If double byte characters are specified in the keyword<x> parameter, this parameter must be set. Additionally, if double byte characters are specified in the keyword<x> or replace<x> parameters, this parameter must be set to string.</p>

BPML	Description
replace<x>	<p>Replacement text that the service should use to replace the keyword. Required. Valid values are hex and string. Default is string.</p> <p><x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: replace1, replace2, replace3, etc.</p> <p>Note: Each replace must have a corresponding keyword. Otherwise, it will be ignored and you will receive an error message.</p> <p>Note: Always define each keyword<x>/replace<x> pair sequentially. For example, do not specify a keyword2/replace2 pair and a keyword4/replace4 pair without a keyword3/replace3 pair between them.</p>
replacetype<x>	<p>Type of replacement text. Optional. Valid values are hex, string, and encoding type for the parameter replace<x>. Encoding type is any supported encoding format such as UTF-8 and UTF-16. Default is string.</p> <p>Note: Encoding is supported only when literal_mode is set to TRUE.</p> <p>The following example demonstrates the use of UTF-8 as the encoding type for the parameter replace<x>:</p> <pre data-bbox="467 768 1252 940"> <output message="DocKeywordReplaceInputMessage"> <assign to="literal_mode">>true</assign> <assign to="keyword1" from "'&#x6E;' "></assign> <assign to="keywordtype1">UTF-8</assign> <assign to="replace1" from="'&#x7F;' "></assign> <assign to="replacetype1">UTF-8</assign> </output> </pre> <p><x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: replacetype1, replacetype2, replacetype3, etc.</p> <p>Note: This parameter is optional in either mode.</p> <p>Note: If double byte characters are specified in the replace<x> parameter, this parameter must be set. Additionally, if double byte characters are specified in the keyword<x> or replace<x> parameters, this parameter must be set to string.</p>
keystart	<p>Characters to add to the beginning of the keyword when the service looks for the string to replace. Optional. Default is \${.</p> <p>Note: Do not specify a value for this parameter if useKeywordDefaults is set to true and you are using String mode.</p> <p>Note: This parameter can also be specified in Application or in the GPM. However, special characters (for example, characters such as parentheses and brackets) <i>must</i> be specified using BPML.</p> <p>Note: This parameter is not used if literal_mode is set to true.</p>
keyend	<p>Characters to add to the end of the keyword when the service looks for the string to replace. Optional. Default is }.</p> <p>Note: Do not specify a value for this parameter if useKeywordDefaults is set to true and you are using String mode.</p> <p>Note: This parameter can also be specified in Application or in the GPM. However, special characters (for example, characters such as parentheses and brackets) <i>must</i> be specified using BPML.</p> <p>Note: This parameter is not used if literal_mode is set to true.</p>

BPML	Description
mode	<p>Specifies the mode to use. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ Document – streams in a complete document in place of a keyword ◆ String – does a simple string replacement <p>Default is String.</p> <p>Note: This parameter is not used if literal_mode is set to true.</p>
useKeywordDefaults	<p>Whether to add the characters specified in keystart and keyend, or their default values, to the start and end of the keyword you are looking for. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Add the characters specified in keystart and keyend, or the default values, to the start and end of the keyword. This is the default setting. ◆ false – Do not add any characters to the start and end of the keyword. <p>Note: This parameter is not used if literal_mode is set to true.</p>
literal_bufferSize	<p>Specifies the size of the buffer used for literal replacements. Optional. Default is 10240 bytes (10 KB).</p> <p>Note: This parameter can also be specified in the Graphical Process Modeler (GPM).</p>
literal_mode	<p>Whether to use high speed literal replacements (literal mode). Valid values: true, false.</p> <p>Note: Literal mode provides high speed performance by limiting processing overhead. You should use this mode unless you specifically need the capabilities of the other modes. The following parameters are not used if literal_mode is set to true:</p> <ul style="list-style-type: none"> ◆ keystart ◆ keyend ◆ mode ◆ useKeywordDefaults <p>Note: This parameter can also be specified in the GPM.</p>
literal_readAheadSize	<p>Specifies the size of the read-ahead buffer used for literal replacements. Optional. Default is 8192 bytes (8 KB). To ensure proper operation, the value specified for this parameter must be smaller than the value specified for the literal_bufferSize parameter.</p> <p>Note: This parameter can also be specified in the GPM.</p>

Choosing a Mode

There are three modes available in the Document Keyword Replace service.

Literal Mode

Because Literal mode operation limits processing overhead and performs the search and replace functions only once, this mode is ideal for high-speed replacement operations. You should always use the Document Keyword Replace service in Literal mode, unless you:

- need to replace a keyword with the contents of a document (use Document mode).
- need recursive replacement capability (use String mode).
- are working with the Synchronization Engine adapters (see *Synchronization Engine Adapters*).

need parameters not available in Literal mode (use String mode).

Note: Because Literal mode limits processing overhead, some parameters are not available in Literal mode. Use String mode if you need these parameters. See *literal_mode* on page 314 for a list of these parameters.

Place the service in Literal mode by setting the **literal_mode** parameter to true. For more information on setting this parameter, see *Configuring the Document Keyword Replace Service* on page 312.

In literal mode, you can also support character deletion.

String Mode

The String mode offers more options and the ability to do recursive replacements. This is the default mode. Place the Document Keyword Replace service in String mode by setting the **mode** parameter to String, or by not specifying a mode. Ensure the **literal_mode** parameter is set to false. For more information on setting this parameter, see *Configuring the Document Keyword Replace Service* on page 312. String mode allows the use of the following parameters:

- keywordtype
- replacetype
- keystart
- keyend
- mode
- useKeywordDefaults

Some situations where you should use String mode include:

- You need the optional parameters available in String mode

- You are working with the Synchronization Engine adapters (see *Synchronization Engine Adapters*)

- You need to do recursive replacements

Note: Because String mode operates recursively, the service contains a mechanism to prevent endless loops when using String mode. You should always test your configuration. If you find that any expected replacements were not made, check your **keyword** and **replace** parameter settings. For example, if you set up the Document Keyword Replace service to replace the string “ABC” with the string “ABCD”, while in String mode, the service will not allow it. Because the **replace** parameter value (ABCD) contains the **keyword** parameter value (ABC), if the service was allowed to run, the result would be an endless series of replacement iterations as shown below:

- ◆ Iteration 1 – ABC
- ◆ Iteration 2 – ABCD
- ◆ Iteration 3 – ABCDD
- ◆ Iteration 4 – ABCDDD
- ◆ Iteration 5 – ABCDDDD
- ◆ This would continue in the same manner endlessly

Document Mode

If you need to replace a string with a complete document, use the Document mode. Place the Document Keyword Replace service in Document mode by setting the **mode** parameter to Document. For more information on setting this parameter, see *Configuring the Document Keyword Replace Service* on page 312.

Business Process Examples

In the following examples, a notation of **#x??** is the hexadecimal equivalent of an unprintable character (?? represents a hexadecimal number).

Note: You cannot just cut and paste (and then run) these examples; you must insert the process name at the beginning of the BPML (such as the following example in which you would replace “somename” with the actual name of the process) and ensure you defined a valid service instance:

```
<process name="somename">
  <!-- within a process, you can have sequences, assigns, and operations -->
</process>
```

Literal Mode

This sample Business Process searches for all line feed characters and replaces them with a carriage return and tilde:

Note: In the example below, **
** is a hex notation is a “character entity reference” that is defined by html (see <http://www.w3.org/TR/REC-html40/sgml/entities.html> for more information).

```
<process name="test">
<sequence>
<operation name="DocKeywordReplaceImpl">
<participant name="myDocKeywordReplaceImplService"/>
<output message="outmsg">
<assign to="." from="*" />
<assign to="literal_mode">true</assign>
<assign to="literal_bufferSize">102400</assign>
<assign to="literal_readAheadSize">8192</assign>
<assign to="keyword1">~</assign>
<assign to="replace1" from="string('&#x0a;')"/>
<assign to="keyword2" from="string('&#x0d;&#x7e;')"/>
<assign to="replace2">Ninety</assign>
</output>
<input message="inmsg">
<assign to="." from="*" />
</input>
</operation>
</sequence>
</process>
```

For the sample business process above, this is the Primary Document before it is run through the Document Keyword Replace service:

```
ISA*00*          *00*          *08*9262390000
*14*9252050234*020129*0525*U*00400*000010006*0*P*:
GS*PD*345345345*9252050234*20020129*0525*10006*T*004010
ST*852*000010874
```

XQ*G*20020128*20020202
XPO*L774211-01
N9*VR*0103103*MEASLES ANIMAL HAVEN
N1*ST*CHART WHSE - LKLD - (942)*9*0069220090942
LIN**UA*003500005221
ZA*QA*594*CA*007*20020128
LIN**UA*003500048553
ZA*QA*594*CA*007*20020128
LIN**UA*003500005537
ZA*QA*576*CA*007*20020128
LIN**UA*003500048571
ZA*QA*800*CA*007*20020128
LIN**UA*003500048501
ZA*QA*600*CA*007*20020128
CTT*5
SE*1*000010874
GE*1*10006
IEA*1*000010006

For the sample business process above, this is the Primary Document after it is run through the Document Keyword Replace service:

ISA*00* *00* *08*9262390000
*14*9252050234*020129*0525*U*00400*000010006*0*P*::~ ß----- Note a Tilde was added
to the end of each line
GS*PD*345345345*9252050234*20020129*0525*10006*T*004010~
ST*852*000010874~
XQ*G*20020128*20020202~
XPO*L774211-01~
N9*VR*0103103*MEASLES ANIMAL HAVEN~
N1*ST*CHART WHSE - LKLD - (942)*9*0069220090942~
LIN**UA*003500005221~
ZA*QA*594*CA*007*20020128~
LIN**UA*003500048553~
ZA*QA*594*CA*007*20020128~
LIN**UA*003500005537~
ZA*QA*576*CA*007*20020128~
LIN**UA*003500048571~
ZA*QA*800*CA*007*20020128~
LIN**UA*003500048501~
ZA*QA*600*CA*007*20020128~
CTT*5~
SE*1*000010874~
GE*1*10006~
IEA*1*000010006~

Document Mode

Note: The path of the replaceX parameter must include a document name from ProcessData. In the following example, **replacementDocument** is the name of the document in ProcessData.

```
<process name="test">
  <sequence>
    <operation name="DocKeywordReplace">
      <participant name="myDocKeywordReplaceService"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="mode">Document</assign>
        <assign to="keyword1">someKeywordToReplace</assign>
        <assign to="replace1">/ProcessData/replacementDocument</assign>
        <assign to="keyword2">anotherKeywordToReplace</assign>
        <assign to="replace2">/ProcessData/PrimaryDocument</assign>
      </output>
      <input message="fromService">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

String Mode

```
<process name="test">
  <sequence>
    <operation name="DocKeywordReplaceImpl">
      <participant name="myDocKeywordReplaceImplService"/>
      <output message="outmsg">
        <assign to="." from="*" />
        <assign to="keyword1">GLN</assign>
        <assign to="replace1">0008586666221</assign>
        <assign to="keyword2">GTIN</assign>
        <assign to="replace2" from="string(myData/GTIN)" />
        <assign to="keyword3">USERNAME</assign>
        <assign to="replace3" from="'user1'" />
        <assign to="keystart" from="'$(' />
        <assign to="keyend" from="'$)'" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Character Deletion Example

Application now supports character deletion as exemplified in the following BPML assignment:

```
<process name="DocKeywordRemoveExample">
  <sequence name="Sequence Start">
    <operation name="Document Keyword Replace">
      <participant name="SomeDocKeywordReplaceInstance"/>
      <output message="DocKeywordReplaceInputMessage">
        <assign to="." from="*" />
        <assign to="literal_mode">true</assign>
        <assign to="literal_bufferSize">102400</assign>
        <assign to="literal_readAheadSize">8192</assign>
        <assign to="keyword1">111111</assign>
        <assign to="replace1" from="string(')'"/>
        <assign to="keyword2">ABC</assign>
        <assign to="replace2" from="string(')'"/>
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Document Tracking Service

The following table provides an overview of the Document Tracking service:

System name	DocumentTracking
Graphical Process Modeler (GPM) category	All Services
Description	Enables you to define correlation name-value pairs to gather additional tracking data about documents.
Business usage	Enables you to gather additional tracking data. You can include one or more tracking services at appropriate points in a business process and define additional tracking data (correlation name-value pairs) to be gathered about a document (or a family of documents).
Usage example	<p>When setting up a business process, you can include one or more tracking services. While checking-in the business process, if document tracking has been enabled, the information discovery process will be responsible for creating the document tracking record (and populate it with an unique tracking ID along with pre-defined tracking data) and the tracking service will add to the contents of the tracking record. However, if document tracking has not been enabled, the tracking service will create the tracking record.</p> <p>When the business process is executed and documents are processed, tracking data about the documents is gathered and passed from one step in the process to the next. If, during one of the steps in the process, a document A gets split into 3 documents, A1, A2 and A3, all 4 documents would share a common tracking ID—the relationships between documents within a family are available when a user searches for documents using tracking information as the search criteria.</p>
Preconfigured?	Yes (DocumentTracking)
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success—The service successfully created or updated a tracking record.◆ Error—The service was not successful in creating or updating a tracking record.
Restrictions	Requires a primary document.

Implementing the Document Tracking Service

To implement the Document Tracking service, complete the following tasks:

1. Create a Document Tracking service configuration. See *Creating a Service Configuration*.
2. Configure the Document Tracking service. See *Configuring the Document Tracking Service* on page 321.
3. Use the Document Tracking service in a business process.

Configuring the Document Tracking Service

To configure the Document Tracking service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration.
use-system-default	The system default is to perform document tracking if it is enabled at the business process level. Optional. Valid values are: <ul style="list-style-type: none">◆ True – Tracking is performed if it is enabled at the business process level.◆ False – Extent of the tracking operation depends on the discovery-level setting. Default.
discovery-level	Optional. Valid values are: <ul style="list-style-type: none">◆ None – Information discovery is not performed. Default.◆ Optional – Information discovery is not performed, if it has not been performed previously.◆ Mandatory – Information discovery is performed, even if it has been performed previously.

Field	Description
tracking-scope	<p>Tracking scope is an advanced option that is used to update system correlations for better control over the system. Optional.</p> <p>Caution: Changing from the default scope (Custom) may cause conflicts with existing correlations. This custom parameter can be added manually through BPML or through the Advanced editor in the Graphical Process Modeler.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ Custom ◆ System ◆ Adapter ◆ EDI ◆ EDIInterchange ◆ EDIGroup ◆ EDITransaction ◆ ACH ◆ CD ◆ MQ ◆ Translation ◆ EDIINT ◆ Mailbox

Business Process Example

The following example illustrates how the Document Tracking service could be used in a business process:

```
<process name="TestDocTracking">
  <sequence>

    <!-- Using custom correlation -->
    <operation name="Tracking">
      <participant name="DocumentTracking"/>
      <output message="msg-to-service">
        <assign to="my-Correlation" from="'myValue'"/>
        <assign to="my-Other-Correlation" from="'myOtherValue'"/>
        <assign to="." from="PrimaryDocument"/>
      </output>
      <input message="msg-from-service">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <!-- Changing system state where needed -->

    <operation name="Tracking">
      <participant name="DocumentTracking"/>
      <output message="msg-to-service">
```

```

    <assign to="State" from="'TESTING'"/>
    <assign to="tracking-scope" from="'SYSTEM'"/>
    <assign to="." from="PrimaryDocument"/>
  </output>
  <input message="msg-from-service">
    <assign to="." from="*"></assign>
  </input>
</operation>

<!-- Do some work here -->

<!-- Update state again -->
<operation name="Tracking">
  <participant name="DocumentTracking"/>
  <output message="msg-to-service">
    <assign to="State" from="'TEST COMPLETE'"/>
    <assign to="tracking-scope" from="'SYSTEM'"/>
    <assign to="." from="PrimaryDocument"/>
  </output>
  <input message="msg-from-service">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</process>

```

Document XPath Replace Service

The Document XPath Replace service enables you to replace the text of a text node in an XML document by specifying an XPath expression.

The following table provides an overview of the Document XPath Replace service:

System name	DocXPathReplace Service
Graphical Process Modeler (GPM) categories	All Services, System, Internet B2B > Transora
Description	The Document XPath Replace service performs text replacements in the document using XPath expressions.
Usage example	Can be used to dynamically replace static text node identifiers with run-time configurable identifiers.
Preconfigured?	An instance of this service is created upon installation. There are no instance configuration variables, but you must configure the workflow variables for the service in the GPM when you are creating a business process.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	By default, any DOCTYPE tag found in the document is removed. If you want to retain the DOCTYPE tag, see <i>How the Document XPath Replace Service Works</i> on page 324.
Returned status values	<ul style="list-style-type: none">◆ Success – Completed successfully.◆ Error – Errors were encountered during text replacement. Check advanced status, status reports, or systems logs.
Restrictions	Must be a well-formed XML document that can be parsed.

How the Document XPath Replace Service Works

The Document XPath Replace service parses a document into a DOM (Document Object Model) so that the specified XPath expression can be evaluated and the resulting node(s) replaced with the specified value(s).

By default, the Document XPath Replace service will use the current primary document unless the *documentKey* parameter is specified to point to a different document. Also by default, and to maintain backward compatibility, the Document XPath Replace service removes any DOCTYPE tag found in the document.

If you need to retain the DOCTYPE tag, there are two ways to do it.

1. The recommended way is to set the *noValidate* parameter to *true* in the BPML. Using this option simply turns off all validation except for "well formed" validation and has the additional benefit of being able to utilize document streaming.
2. The other option is to set the *keepDocType* parameter to *true* in the BPML. Using this option will still remove the DOCTYPE tag before parsing, but will re-add it before returning the updated document.

Starting with Application version 4.0, the Document XPath Replace service also has the ability to perform multiple (batch) replacements with just one call to the adapter instead of separate individual calls.

Implementing the Document XPath Replace Service

To implement the Document XPath Replace service, simply add the existing instance to your business process using the GPM and configure the workflow parameters appropriately.

Configuring the Document XPath Replace Service

To configure the Document XPath Replace service, you must specify the following field settings (workflow parameters) in the GPM:

Field	Description
Config	Name of the service configuration. Required.
debug	Turns on debugging for this workflow which logs extra messages to the system log. Optional. Valid values are Yes (true) and No (false). Default is No (false).
documentKey	Name of the document in the workflow context in which to replace text. Optional. If no document name is specified in this parameter, the primary document is used.
keepDocEncoding	Allows you to maintain the original document encoding for your XML documents to your generated document. For example, if you use an XML document with an encoding attribute of <code><?xml version= '1.0' encoding= 'UTF-16' ?></code> , you can keep this encoding attribute value for your generated document by selecting Yes in the GPM for keepDocEncoding or by adding the following line if you are editing the BPML, <code><assign to="keepDocEncoding">true</assign></code> . After the Document XPath Replace service runs, the original header of <code><?xml version= '1.0' encoding= 'UTF-16' ?></code> will be the header for the generated document. If you do not provide this parameter, the document will be encoded using "UTF-8" as the default value and a header of <code><?xml version= '1.0'></code> will be given to the generated document. Valid values are Yes (true) and No (false). Optional.

Field	Description
keepDocType	Maintains the DOCTYPE tag by stripping the tag before performing any XPath replacements and then re-adding it back to the document before returning. Large file support cannot be utilized with this parameter. Optional. Valid values are Yes (true) and No (false). Default is No (false).
noValidate	Disables any parser validation to keep from removing any DOCTYPE tags. This field overrides any use of the 'keepDocType' parameter. Using this field has the additional benefit of using document streaming. Optional. Valid values are Yes (true) and No (false). Default is No (false).
prefix	Prefix used with a unique identifier as the replacement text. If performing more than one replacement (batch), this field must be sequentially numbered to match up with the corresponding textNodeXPath field. Optional, but either prefix or replacementText must be specified.
replacementText	Text to replace in the document. If performing more than one replacement (batch), this field must be sequentially numbered to match up with the corresponding textNodeXPath field. Optional, but either prefix or replacementText must be specified.
replaceMultiple	Whether multiple occurrences of the XPath statement should be replaced or just the first it finds. If performing more than one replacement (batch), this field must be sequentially numbered to match up with the corresponding textNodeXPath field. Optional. Valid values are Yes (true) and No (false). Default is No (false).
textNodeXPath	XPath identifying the text node whose value needs to be replaced. Required.

Business Process Examples

This example performs a single text replacement.

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*" />
    <assign to="textNodeXPath">//some/tag/text()</assign>
    <assign to="replacementText" from="'new text'"/>
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
```

This example performs a single text replacement, but does it for multiple occurrences:

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath" from="//some/tag/text()" />
    <assign to="replacementText" from="'new text'"/>
    <assign to="replaceMultiple">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs a single text replacement, but does it for multiple occurrences and generates a unique identifier with the supplied prefix of CMD- as the replacement text.

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath" from="//some/tag/text()" />
    <assign to="prefix">CMD-</assign>
    <assign to="replaceMultiple">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs a single text replacement and one multiple occurrence text replacement:

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath1" from="//some/tag/text()" />
    <assign to="replacementText1" from="'new text'"/>
    <assign to="textNodeXPath2" from="//some/trans/tag/text()" />
    <assign to="prefix2">TRANS-</assign>
    <assign to="replaceMultiple2">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs two different text replacements, both with multiple occurrences and both generating unique identifiers with the corresponding prefix as the replacement text:

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath1" from="//some/cmd/tag/text()" />
    <assign to="prefix1">CMD-</assign>
    <assign to="replaceMultiple1">true</assign>
    <assign to="textNodeXPath2" from="//some/trans/tag/text()" />
    <assign to="prefix2">TRANS-</assign>
    <assign to="replaceMultiple2">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs eight different text replacements and uses the noValidate parameter, which keeps the parser from validating any schema or DTD and does not remove the DOCTYPE tag (if any):

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
```

```

<assign to="." from="*" />
<assign to="noValidate">true</assign>
<!-- Update the primary document with current year -->
<assign to="textNodeXPath1">//CNTROLAREA/DATETIME/YEAR/text()</assign>
<assign to="replacementText1" from="substring(formattedTime, '1', '4')"/>
<!-- Update the primary document with current month -->
<assign to="textNodeXPath2">//CNTROLAREA/DATETIME/MONTH/text()</assign>
<assign to="replacementText2" from="substring(formattedTime, '5', '2')"/>
<!-- Update the primary document with current day -->
<assign to="textNodeXPath3">//CNTROLAREA/DATETIME/DAY/text()</assign>
<assign to="replacementText3" from="substring(formattedTime, '7', '2')"/>
<!-- Update the primary document with current hour -->
<assign to="textNodeXPath4">//CNTROLAREA/DATETIME/HOUR/text()</assign>
<assign to="replacementText4" from="substring(formattedTime, '9', '2')"/>
<!-- Update the primary document with current minute -->
<assign to="textNodeXPath5">//CNTROLAREA/DATETIME/MINUTE/text()</assign>
<assign to="replacementText5" from="substring(formattedTime, '11', '2')"/>
<!-- Update the primary document with current second -->
<assign to="textNodeXPath6">//CNTROLAREA/DATETIME/SECOND/text()</assign>
<assign to="replacementText6" from="substring(formattedTime, '13', '2')"/>
<!-- Update the primary document with current subsecond -->
<assign to="textNodeXPath7">//CNTROLAREA/DATETIME/SUBSECOND/text()</assign>
<assign to="replacementText7" from="substring(formattedTime, '15', '4')"/>
<!-- Update the primary document with current timezone -->
<assign to="textNodeXPath8">//CNTROLAREA/DATETIME/TIMEZONE/text()</assign>
<assign to="replacementText8" from="timezoneOffsetFromGMT/text()"/>
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>

```

Dynamic Services

Dynamic services enable Application to consume Web services as it does other services – as part of a business process. Services are created based on the WSDL that you check in to Application. These services can then communicate with a specified web service using SOAP messages as part of a business process, enabling you to extend your business to use external web functionality.

The following table provides an overview of dynamic services.

System Name	Determined by the information in the WSDL.
Graphical Process Modeler (GPM) categories	Dynamic Services
Description	Enables Application to consume a web service as a normal service. User can check in the WSDL of the consumed web service into Application. Once checked in, Application creates service definition and service instances corresponding to operations within the WSDL and adds those service instances in the GPM Palette. User can then use these services as a normal service and build a BPML out of it. Once this BPML get executed these services will be called, which in turn will make a SOAP call to the service endpoint and return the response back to the BPML.
Business usage	Include web service functionality in a business process.
Usage example	You need to do currency conversions in a business process, and are aware of a web service that performs conversions on the fly. You can have a dynamic service to include in a business process. The service sends an amount and currency code to the web service, along with the code of the currency to convert to, and receives the converted value back from the web service.
Preconfigured?	No. A dynamic service is created by checking in a WSDL file.
Requires third party files?	WSDL, and if required for signing SOAP messages, certificates and security tokens
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ SOA Inbound service (This is an internal service – no configuration required. It is configured by Web Services.)◆ SOA Outbound service (This is an internal service – no configuration required. It is configured by Web Services.)◆ HTTP Client adapter (There is no configuration required for this adapter. It is configured by Web Services.)
Application requirements	Application Web Services
Initiates business processes?	No. Dynamic services are used within a business process.
Invocation	As part of a business process
Business process context considerations	Requires that values for parameters are passed through workflow context to the service; once web service returns a result, the result becomes the primary document.

Returned status values	Success, Error
Restrictions	None
Persistence level	System default
Testing considerations	To test a dynamic service, you must include it in a business process and run the process. For an overview of the procedure to follow, see <i>Implementing a Dynamic Service</i> on page 331.

How Dynamic Services Are Created

Application creates dynamic services based on information provided in a WSDL file that you check in.

In the WSDL, you specify each web service and port combination that is required. Application creates a GPM stencil category for each unique web service and port combination.

The name of the stencil category will always be in upper case to avoid any conflicts with operating system restrictions. Therefore, you cannot specify two categories with the same name but with different capitalization.

In the WSDL, you also specify any operations needed for each web service and port. Application creates a service definition and configuration of the service definition for each operation on the web service and port.

Once created, the services are displayed in the GPM and can be used in business processes.

Note: Only SOAP bindings are considered for dynamic services. Other types of bindings are ignored and dynamic services are not created for them.

Once the Dynamic Service creation process is done, you can begin using the services in your business processes. The dynamically created services can only be used as part of a business process.

What Happens During a Business Process

When a business process is invoked, the dynamic service loads the WSDL (which has been stored in a cache by the check in process), creates a SOAP request for the specified operation, and posts it to the endpoint specified in the WSDL. The dynamic service then waits for the response from the service endpoint. Once the endpoint returns a response, the dynamic service loads the response as the primary document, and into the work flow context, so that next service in the business process can take appropriate action on the returned response.

Example

Your corporation has a web service that performs currency conversions. You create a business process that receives values in U.S. dollars, but needs to have the amounts converted to Euros for a later step in the process.

You define WSDL for the web service, specifying the information necessary to dynamically create a service, and check in the WSDL.

Application validates the WSDL and creates the new service and a stencil and category for it in the GPM. It adds the service definition to the list of services available in Application for creation, editing, and copying. It creates as many configurations of the service as needed for the functions you specify in the WSDL. So, if

you specified “Convert” as the web service to go to in the WSDL and “US_to_Euro” and “Euro_to_Yen” as the actions needed, a service category called Convert would be created on the Dynamic Services stencil. Within that new category, one service definition and one service configuration would be created for each action – US_to_Euro and Euro_to_Yen.

You include the US_to_Euro service in your business process. The US_to_Euro service sends a SOAP message containing the value (the U.S. dollar amount, say \$50) to the Convert web service. The web service does the conversion to Euros, and sends back the converted amount to the US_to_Euro service, which passes the new Euro amount to the primary document. The next step in the business process uses the Euro amount for its operation.

You must have Application Web Services to use this functionality. You create a WSDL file that provides information about the service (or services), and Application creates the services and makes them available in the GPM for you to configure.

Implementing a Dynamic Service

To implement a dynamic service, complete the following tasks:

1. Obtain the WSDL for the web service you want to access during a business process.
2. Verify that the WSDL contains the necessary content to describe the dynamic service to Application. If it does not, you must edit the WSDL to include the required information. See *About WSDL* on page 344.
3. Check in the WSDL.
4. In the GPM, create your business process and select the desired service from the newly created stencil, then select a configuration of the service.
5. Specify the parameter settings for the service configuration.
6. Check in, test, and use the business process.

Configuring a Dynamic Service

You must specify field settings in Application, using the GPM.

Setting Up the Service in the GPM

The four security-related parameters in the table (Certificate, InsertSecurityHeader, SecurityToken, X509CertificateOption) are always displayed in the GPM for all dynamic services.

Additionally, a dynamic service can have parameters based on the partType element in the WSDL. If the partType element is from the XML Schema namespace (simple type), then these parameters will also be displayed in GPM. This normally happens for the RPC/Literal style WSDL. However, if the partType of

these parameters refers to another namespace, then these parameters are not displayed in the GPM service editor. This normally happens for the Document/Literal style WSDL.

Field	Description
Config	Select the name of the service configuration from the list.
Certificate	Enter the name of the digital certificate to use for signing the SOAP request. This certificate must have already been imported to the Web service endpoint system.
InsertSecurityHeader	Whether to insert the security header. Valid values are true and false.
SecurityToken	Name of the security token. This token must have already been imported to the Web service endpoint system.
X509CertificateOption	Used when inserting a digital certificate into the SOAP request. Valid values are: <ul style="list-style-type: none"> ◆ Binary Token ◆ IssuerSerial (default) ◆ X509KeyIdentifier

Business Process Example – RPC/Literal

The following example illustrates how a dynamic service could be used in a business process:

You need to access a web service that does simple mathematical calculations during a business process step. The web service is called Calculator. The WSDL for the Calculator is shown in the following example.

1. Check in the WSDL to Application.
2. Create a business process and include one or more configurations of the new service in it.
3. Check in the business process.
4. Test the dynamic service by running the business process. If there is a problem accessing the web service, error messages are displayed in the BP Monitor.

The following example shows a WSDL file with the name Calculator.

The following example shows creating a dynamic service creation against a WSDL with one port type and with two operations:

```
<?xml version="1.0" encoding="UTF-8"?>
  <wsdl:definitions name="CalculatorWSDL"
targetNamespace="http://00.00.00.00:1234/axis/Calculator.jws"
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://00.00.00.00:1234/axis/Calculator.jws"
xmlns:intf="http://00.00.00.00:1234/axis/Calculator.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
xmlns:wSDLsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:message name="subtractResponse">
    <wsdl:part name="subtractReturn" type="xsd:int"/>
  </wsdl:message>
  <wsdl:message name="addResponse">
```

```

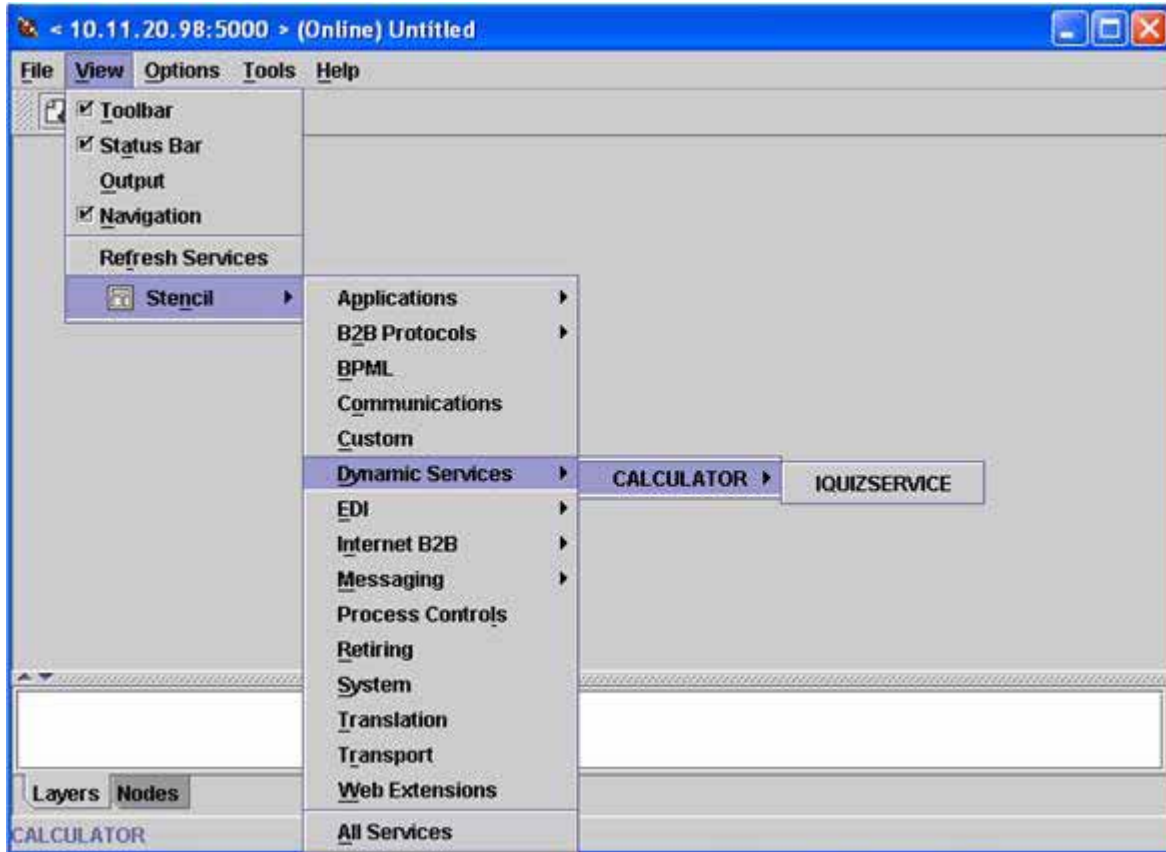
        <wsdl:part name="addReturn" type="xsd:int"/>
    </wsdl:message>
    <wsdl:message name="addRequest">
        <wsdl:part name="i1" type="xsd:int"/>
        <wsdl:part name="i2" type="xsd:int"/>
    </wsdl:message>
    <wsdl:message name="subtractRequest">
        <wsdl:part name="i1" type="xsd:int"/>
        <wsdl:part name="i2" type="xsd:int"/>
    </wsdl:message>
    <wsdl:portType name="Calculator">
        <wsdl:operation name="add" parameterOrder="i1 i2">
            <wsdl:input message="intf:addRequest" name="addRequest"/><wsdl:output
message="intf:addResponse" name="addResponse"/>
        </wsdl:operation>
        <wsdl:operation name="subtract" parameterOrder="i1 i2">
            <wsdl:input message="intf:subtractRequest" name="subtractRequest"/>
            <wsdl:output message="intf:subtractResponse" name="subtractResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="CalculatorSoapBinding" type="intf:Calculator">
        <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="add">
            <wsdlsoap:operation soapAction=""/>
            <wsdl:input name="addRequest">
                <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
            </wsdl:input>
            <wsdl:output name="addResponse">
                <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://00.00.00.00:1234/axis/Calculator.jws" use="encoded"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="subtract">
            <wsdlsoap:operation soapAction="foo"/>
            <wsdl:input name="subtractRequest">
                <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://DefaultNamespace" use="encoded"/>
            </wsdl:input>
            <wsdl:output name="subtractResponse">
                <wsdlsoap:body
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="http://00.00.00.00:1234/axis/Calculator.jws" use="encoded"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="CalculatorService">
        <wsdl:port binding="intf:CalculatorSoapBinding" name="IQuizService">
            <wsdlsoap:address location="http://00.00.00.00:1234/axis/Calculator.jws"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>

```

When the file is checked in, Application attempts to validate this WSDL file and on successful validation, it creates services in the GPM stencil:

Dynamic Services > CALCULATOR (name of the WSDL file)

Dynamic Services > CALCULATOR > IQUIZSERVICE (Port Name)



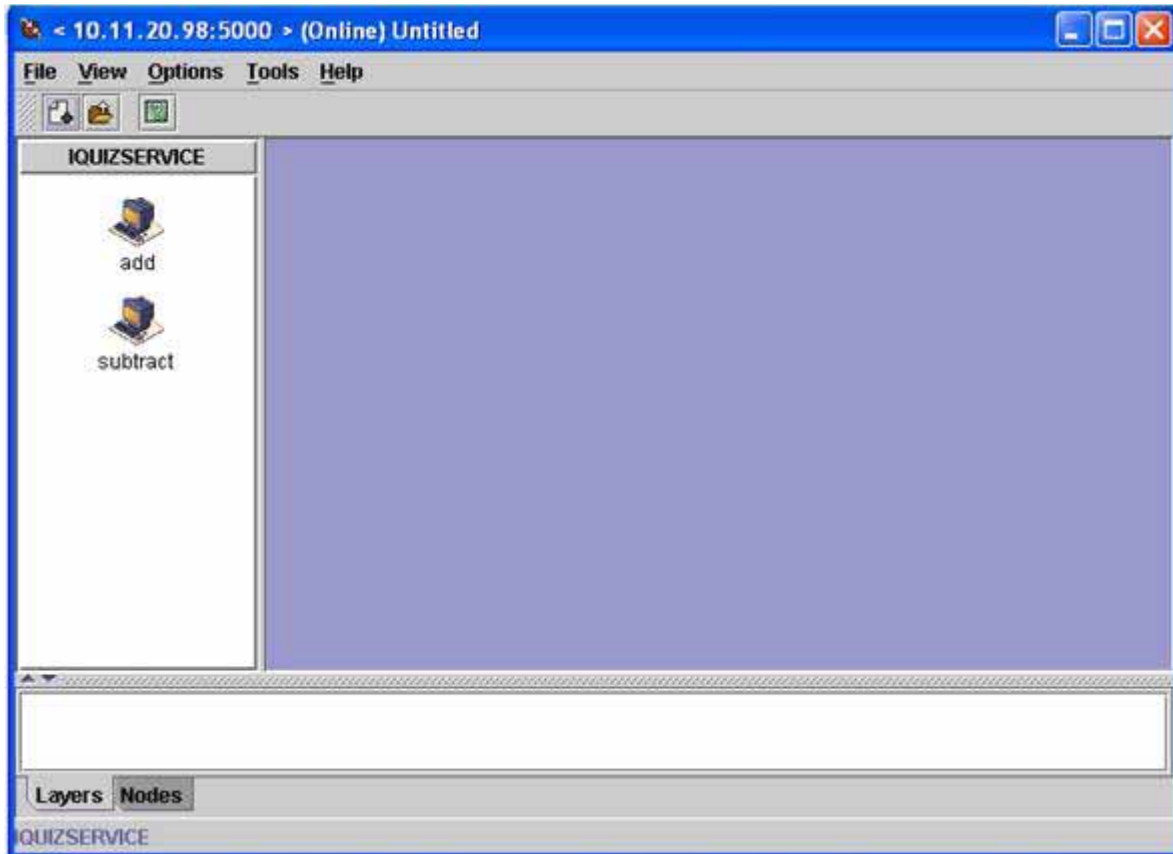
For each operation listed under a particular port, a service definition file is created in Application. For example, for the two operations (add and subtract) in the WSDL shown in the previous example, the following two service definitions file get created:

CALCULATOR_add

CALCULATOR_subtract

A new stencil will open which will then display all the operations this port type will have as a Application service.

The add and subtract operations are displayed as services for the IQUIZSERVICE stencil in the GPM, as shown in the following example:



Using a Dynamic Service in a Business Process

Once a dynamic service is created, you can include it in a business process, as shown in the following example. Note that the add service icon is selected. In the service editor, the default service instance

(DS_CALCULATOR_PORT1_OPE2_Instance) has been selected, and its parameters are displayed in the editor in the lower half of the screen. Values have been assigned for the i1 and i2 part name parameters

Name	Value	Use XPATH?	Append?
Certificate		<input type="checkbox"/>	<input type="checkbox"/>
i1	20	<input type="checkbox"/>	<input type="checkbox"/>
i2	30	<input type="checkbox"/>	<input type="checkbox"/>
InsertSecurityHeader		<input type="checkbox"/>	<input type="checkbox"/>
SecurityToken		<input type="checkbox"/>	<input type="checkbox"/>
X509CertificateOption		<input type="checkbox"/>	<input type="checkbox"/>

The Certificate, InsertSecurityHeader, SecurityToken and X509CertificateOption parameters appear in every dynamic service configuration, but are only used when signing the SOAP request. The remaining two parameters, i1 and i2, are specific to this service. These are the part names of the input message of the add operation of the Calculator WSDL. Values have been entered for them. (Normally, these values would be passed to the service by an earlier step in the business process.)

Note: The parameters for a dynamic service are displayed in the GPM only when the partType is from the XML schema namespace (simple type). If the partType refers to another namespace, the parameters do not display in the GPM and it must be passed in to the service as a document.

Validate, save, and check in the business process to Application. When the business process runs, the add service is invoked, reads the WSDL (which has been stored in a cache by the check in process), generates a SOAP message, sends it to the Web service, and waits for a response. The Web service returns the response, it becomes the primary document (and therefore available to subsequent steps in the business process), and the business process continues.

Request by Dynamic Service to Web Service

The following example shows a request:

```
<soapenv:Envelope xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:tns0="http://DefaultNamespace"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body
    soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
    <tns0:add>
      <i1 xsi:type="xs:int">20</i1>
```



```

        <i2 xsi:type="xs:int">30</i2>
    </tns0:add>
</soapenv:Body>
</soapenv:Envelope>

```

Response from the Web Service

The following is the response received by the add service from the web service:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <ns1:addResponse
            soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:ns1="http://DefaultNamespace">
            <addReturn xsi:type="xsd:int">50</addReturn>
        </ns1:addResponse>
    </soapenv:Body>
</soapenv:Envelope>

```

Errors Generated

If any input was invalid, or if there is an internal processing error at the web service endpoint, a SOAP_FAULT will be returned. The following is an example of a SOAP fault response for invalid input obtained from the web service endpoint:

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
        <soapenv:Fault>
            <faultcode>soapenv:Server.userException</faultcode>
            <faultstring>java.lang.NumberFormatException: z20</faultstring>
            <detail>
                <ns1:hostname
                    xmlns:ns1="http://xml.apache.org/axis/">ADT</ns1:hostname>
            </detail>
        </soapenv:Fault>
    </soapenv:Body>
</soapenv:Envelope>

```

If the endpoint specified in the WSDL is not running, then after the waiting time specified in the HTTP Client adapter, the business process at the client end will return the error NO RESPONSE FROM ENDPOINT.

Dynamic Service for Document/Literal style WSDL

The following is an example of a Document/Literal style WSDL with one portType and one operation:

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions name="BeadInquiryWS"
    targetNamespace="http://www.sterlingcommerce.com/mesa"
    xmlns:mesa="http://www.sterlingcommerce.com/mesa"
    xmlns:mesa_xsd="http://www.sterlingcommerce.com/mesa/schema"
    xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
    xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"

```

```

xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
<wSDL:types>
  <xs:schema attributeFormDefault="unqualified"
    elementFormDefault="qualified"
    targetNamespace="http://www.sterlingcommerce.com/mesa/schema"
    xmlns="http://www.sterlingcommerce.com/mesa/schema"
    xmlns:tns="http://www.sterlingcommerce.com/mesa/schema"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
  <xs:complexType name="Binary">
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="href" type="xs:anyURI"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:element name="attachment" type="tns:Binary"/>
  <xs:element name="inlineAttachment" type="xs:base64Binary"/>
  <xs:complexType name="ProcessData">
    <xs:sequence>
      <xs:any/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="mesaFault" type="tns:MESAFault"/>
  <xs:complexType name="MESAFault">
    <xs:sequence>
      <xs:element name="code"/>
      <xs:element name="message"/>
      <xs:element name="statusReport"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="processData" type="tns:ProcessData"/>
  <xs:element name="documents">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="tns:attachment"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:simpleType name="HashType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="MD5"/>
      <xs:enumeration value="NONE"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="MESAAuth">
    <xs:sequence>
      <xs:element name="principal"/>
      <xs:element name="auth">
        <xs:complexType>
          <xs:simpleContent>
            <xs:extension base="xs:string">
              <xs:attribute name="hashType" type="tns:HashType"/>
            </xs:extension>
          </xs:simpleContent>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
    <xs:element name="mesaAuth" type="tns:MESAAuth"/>
    <xs:element name="Bead_Inquiry" type="tns:ProcessData"
xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
</xs:schema>
</wsdl:types>
<wsdl:message name="MESAResponse">
    <wsdl:documentation xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
        <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
    </wsdl:documentation>
    <wsdl:part element="mesa_xsd:processData" name="parameters"/>
    <wsdl:part element="mesa_xsd:attachment" name="attachment"/>
</wsdl:message>
<wsdl:message name="Bead_Inquiry">
    <wsdl:documentation xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
        <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
    </wsdl:documentation>
    <wsdl:part element="mesa_xsd:mesaAuth" name="header"/>
    <wsdl:part element="mesa_xsd:Bead_Inquiry" name="parameters"/>
</wsdl:message>
<wsdl:message name="MESAFault">
    <wsdl:documentation xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
        <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
    </wsdl:documentation>
    <wsdl:part element="mesa_xsd:mesaFault" name="parameters"/>
</wsdl:message>
<wsdl:message name="GISGeneric">
    <wsdl:documentation xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
        <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
    </wsdl:documentation>
    <wsdl:part element="mesa_xsd:mesaAuth" name="header"/>
    <wsdl:part element="mesa_xsd:processData" name="parameters"/>
</wsdl:message>
<wsdl:portType name="GISPortType">
    <wsdl:documentation xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
        <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
    </wsdl:documentation>
    <wsdl:operation name="executeBead_Inquiry">
        <wsdl:documentation
xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
            <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
        </wsdl:documentation>
        <wsdl:input message="mesa:Bead_Inquiry"/>
        <wsdl:output message="mesa:MESAResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="GISBinding" type="mesa:GISPortType">
    <wsdl:documentation xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
        <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
    </wsdl:documentation>
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="executeBead_Inquiry">
        <soap:operation soapAction="sii:Bead_Inquiry"/>
    <wsdl:input>

```

```

        <soap:body parts="parameters header" use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <mime:multipartRelated>
            <mime:part>
                <soap:body parts="parameters" use="literal"/>
            </mime:part>
            <mime:part>
                <mime:content part="attachment"
type="application/octetstream"/>
            </mime:part>
        </mime:multipartRelated>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="BeadInquiryWS">
    <wsdl:port binding="mesa:GISBinding" name="GISPort">
        <soap:address
location="http://10.11.20.98:5040/soap?service=BeadInquiryWS"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

Once the file is checked in and validated, the services are created in the GPM stencil. This WSDL file, beadInquiry, would have the following entries in the GPM:

Dynamic Services > BEADINQUIRY (name of the WSDL file)

Dynamic Services > BEADINQUIRY > GISPORT (Port Name)

Application creates a new service definition called BEADINQUIRY_executeBead_Inquiry and a default service instance called DS_BEADINQUIRY_PORT1_OPE1_Instance.

The main difference between execution of an RPC/Literal based dynamic service and a Document/Literal based dynamic service is that the Document/Literal requires many documents to be present in the workflow context of the executed business process. For example, to execute the executeBead_Inquiry service, the workflow context of the executed business process should contain the following two documents, one each for the mesaAuth and Bead_Inquiry element:

MesaAuth.txt:

```

<mesa:mesaAuth xmlns:mesa="http://www.sterlingcommerce.com/mesa">
    <principal>admin</principal>
    <auth hashType="NONE">password</auth>
</mesa:mesaAuth>

```

BeadInquiry.txt:

```

<mesa:Bead_Inquiry xmlns:mesa="http://www.sterlingcommerce.com/mesa"/>

```

Both elements are part of the input message of the executeBead_Inquiry operation of the checked-in WSDL file – the number and type of these required documents depends on the input message of the operation.

Before execution of any document/literal based dynamic service, all documents mentioned in the input message of the operation should be present in the workflow context.

How these documents will be available in the workflow context depends on how you write the business process. You can write some business specific services or adapters, or can use some system level services or adapters to put these prerequisite documents into the workflow context. For example, to execute the `executeBead_Inquiry` service, this example uses the File System adapter to populate the two documents in the workflow context.

The first File System adapter collects the `BeadInquiry.txt` file from the file system and the second collects the `mesaAuth.txt` file from the file system. Once these two documents are in the workflow context, you can extract the two documents from the primary document and assign them to `Bead_Inquiry` and `mesaAuth` elements using the two Assign services, as shown in the following example.



The following BPML example shows how the business process might look:

```
<process name="default">
  <sequence>
    <operation name="File System Adapter">
      <participant name="AS2Extract"/>
      <output message="FileSystemInputMessage">
        <assign
to="attachFile"/>/server1/share/joeuser/20060307/BeadInquiry.txt</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <assign to="Bead_Inquiry" from="//PrimaryDocument/@SCIOBJECTID"></assign>
    <operation name="File System Adapter">
      <participant name="AS2Extract"/>
      <output message="FileSystemInputMessage">
        <assign to="attachFile"/>/server1/joeuser/20060307/MesaAuth.txt</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <assign to="mesaAuth" from="//PrimaryDocument/@SCIOBJECTID"></assign>
    <operation name="executeBead_Inquiry">
      <participant name="DS_BEADINQUIRY_PORT1_OPE1_Instance"/>
      <output message="DS_BEADINQUIRY_PORT1_OPE1InputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

```
</process>
```

Because we are not generating a signed SOAP request, leave the four parameters blank.

Validate, save, check in, and execute the business process. When executed, the following happens:

The executeBead_Inquiry service reads the WSDL file, identifying the operation and the input message that need to be executed.

For each part of the input message, the service tries to obtain the corresponding document from the workflow context. For this example, the executeBead_Inquiry service will try to obtain the documents named mesaAuth and Bead_Inquiry. If the service fails to obtain these two documents, the service throws an exception that it cannot find the specified document in the process data. If it succeeds, the service generates the SOAP message, sends it to the endpoint, and waits for the response.

Request Sent to Web Service

The following example shows the SOAP message generated by the executeBead_Inquiry service:

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
  <mesa:Bead_Inquiry
    xmlns:mesa="http://www.sterlingcommerce.com/mesa">
  </mesa:Bead_Inquiry>
  <mesa:mesaAuth xmlns:mesa="http://www.sterlingcommerce.com/mesa">
    <principal>admin</principal>
    <auth hashType="NONE">password</auth>
  </mesa:mesaAuth>
  </soapenv:Body>
</soapenv:Envelope>
```

Response from Web Service

The following example shows the response the executeBead_Inquiry service received from the web service endpoint:

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
  <soapenv:Body>
  <mesa:processData
    xmlns:mesa="http://www.sterlingcommerce.com/mesa">
  <ProcessData>
    <service>BeadInquiryWS</service>
    <b2b-protocol>http</b2b-protocol>
    <transport-instance-id>MESAHttpServerAdapter_HttpServerAdapter_node1
  </transport-instance-id>
    <http-request-uri>/soap</http-request-uri>
    <transport-session-id>Mon Mar 13 11:33:49 IST 2006:26</transport-session-id>
    <messageMode>1</messageMode>
    <wsConfig name="BeadInquiryWS">
      <certID></certID>
      <verificationCertID></verificationCertID>
    </wsConfig>
  </ProcessData>
  </mesa:processData>
  </soapenv:Body>
</soapenv:Envelope>
```

```

    </wsConfig>
    <SOARequiredSignature>>false</SOARequiredSignature>
    <EXPECT_SECURITY_HEADER>>false</EXPECT_SECURITY_HEADER>
    <SOAP_URI>/soap</SOAP_URI>
    <SOAPEnvNSPrefix>soapenv</SOAPEnvNSPrefix>
  <SOAPEnvNSURI>
http://schemas.xmlsoap.org/soap/envelope/
  </SOAPEnvNSURI>
  <mesa xmlns="uri:sci">
    <Bead_Inquiry
      xmlns="http://www.sterlingcommerce.com/mesa"></Bead_Inquiry>
  </mesa>
    <serviceMode>0</serviceMode>
    <typeName>Bead_Inquiry</typeName>
    <PrimaryDocument SCIOBJECTID="blrgislin01:3399d4b3:109f23357e5:-4260">
    </PrimaryDocument>
    <ADD_SOAP_ENVELOPING>>false</ADD_SOAP_ENVELOPING>
    <SOAPOutboundAttachments>
      <SOAPAttachment1
        Content-ID="cid:attachment=
          1015796913_1142248458099@sterlingcommerce.com"
        SCIOBJECTID="blrgislin01:3399d4b3:109f23357e5:-4245">
      </SOAPAttachment1>
    </SOAPOutboundAttachments>
    <attachmentCID>
      cid:attachment=1015796913_1142248458099@sterlingcommerce.com
    </attachmentCID>
    <INSERT_SECURITY_HEADER>>false</INSERT_SECURITY_HEADER>
  </ProcessData>
</mesa:processData>
<mesa:attachment
  xmlns:mesa="http://www.sterlingcommerce.com/mesa"
href="cid:attachment=1015796913_1142248458099@sterlingcommerce.com">
</mesa:attachment>
</soapenv:Body>
</soapenv:Envelope>

--_29258051243737204Sterling29258051243737204MOKO
content-type: application/octet-stream
content-id: <attachment=1015796913_1142248458099@sterlingcommerce.com>

<?xml version='1.0' encoding='UTF-8'?>
  <result><row><OBJECT_ID>B2B_WF_OBJECT_ID_2</OBJECT_ID>
  <OBJECT_VERSION>1.0</OBJECT_VERSION>
  <OBJECT_NAME>HTTP_SEND_ENVELOPE_OFF</OBJECT_NAME>
  <TRANSPORT_DESC>HTTP</TRANSPORT_DESC>
  <ENVELOPE_DESC>NO- ENVELOPE
</ENVELOPE_DESC>
<MESSAGE_MODE>send</MESSAGE_MODE>
<WORKFLOW_VERSION>1.0</WORKFLOW_VERSION>
<OBJECT_CLASS>B2B_WORKFLOW</OBJECT_CLASS>
<LAST_MODIFICATION></LAST_MODIFICATION>
<LAST_MODIFIER></LAST_MODIFIER><OBJECT_STATE></OBJECT_STATE></row>
--_29258051243737204Sterling29258051243737204MOKO--

```

About WSDL

For creation of a dynamic service to succeed, the following elements need to be included in the WSDL:

Note: See *WSDL Validation* for more information about specific rules used the validation process.

Element	Description
Service	At least one service with a name associated with it.
Ports	For each service, there should be one or more ports, each with a unique name.
SOAP port	At least one port must have a SOAP address location specified for it.
Endpoint	Each SOAP address must include a location attribute that specifies the endpoint where the web service runs. The location must be a proper URI and the protocol must be HTTP.
Binding	<p>Each port must refer to a binding element, using the binding attribute. For dynamic services, only SOAP binding over HTTP protocol is supported. If the WSDL contains both SOAP and non-SOAP bindings, dynamic services are created for only the SOAP addresses – non-SOAP bindings are ignored.</p> <p>The binding element describes how the service is bound to a messaging protocol, particularly the SOAP messaging protocol. You can use either Remote Procedure Call (RPC) or Document style bindings. If the attribute is not present, then the system assumes the style is Document. The SOAP binding can also have an encoded use or a literal use.</p>
Operations	The number of operations in the binding section and the port type section must be the same. The names must be the same.
Input/Output Messages in Operations	Each operation must have input and output operations associated with it. The Dynamic Service Creation supports only Request-Response transmission and does not support solicit-response or one-way transmission types.
Parts	Each message can include zero or more parts. Each part must have a unique name.
Part Element	If the WSDL uses Document/Literal style binding, each part must have a part element that refers to a root element of the schema defined in the WSDL.
Duplicate Operations	WSDL files that use RPC binding can have two duplicate operations, but not more than that. WSDL that uses Document/Literal binding can never have any duplicate operations.

Checking In a New WSDL

If you check in a new version of a WSDL file, the following happens:

- Deletion of old dynamic services based on the old WSDL.

- Creates new dynamic services based on the new WSDL file

Note: When a WSDL has multiple versions, the dynamic service will always be created for the default version only.

Deleting Dynamic Services

To delete a dynamic service, delete the WSDL. When you delete a checked-in WSDL file, Application deletes the following items, in this sequence:

1. All service configurations associated with the WSDL.
2. All service definitions associated with the WSDL.
3. The XML file created for the service definition.
4. GPM stencil categories and items associated with the WSDL.

E-5 2000 Adapter

The E-5 2000 adapter enables business processes to interact with E-5 clients and servers. E-5 is a B2B standard developed by the Automotive Industry Action Group (AIAG). This adapter provides computer-to-computer delivery and receipt of EDI, XML, or other proprietary data agreed upon by the trading partners. The E-5 2000 adapter has two primary purposes:

- Provide E-5 client-side services

- Provide E-5 server-side services

The E-5 2000 adapter uses predefined business processes to assist communication with other E-5 servers or clients. These business processes provide client and server functions for delivering a document, obtaining a resultant document, and acknowledging receipt of a resultant document.

The following table provides an overview of the E5 Client Configuration adapter:

System name	None
Graphical Process Modeler (GPM) categories	All Services, Internet B2B
Description	The E5 Client Configuration adapter is a set of model business processes that allows a user to send and retrieve documents from an E-5 server.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	E5 Client Configuration service
Application requirements	The adapter supports E-5 2000 or version 2.0 of the E-5 standard. Previous versions of E-5 are not supported.
Initiates business processes?	No
Invocation	The API Definition, Deliver Submit, Obtain Submit and Acknowledge Submit run by starting the E5Client_StartSubmit.bpml. The Loop Back Test runs by starting the E5Client_LoopBackTest.bpml

Business process context considerations	<ul style="list-style-type: none"> ◆ API Definition – Run E5ClientConfig. No documents in ProcessData. ◆ Deliver Submit – Run E5ClientConfig. The Deliver Submit document should be placed as PrimaryDocument and attachment document as AttachmentDocument in ProcessData. ◆ Obtain Submit – Run E5ClientConfig. The Obtain Submit document should be placed as PrimaryDocument in ProcessData. ◆ Acknowledge Submit – Run E5ClientConfig. The Acknowledge Submit document should be placed as PrimaryDocument in ProcessData. ◆ Loop Back Test – Run E5ClientConfig. The Deliver Submit document should be placed as PrimaryDocument and attachment document as AttachmentDocument in ProcessData.
Returned status values	If a message code of Serious is returned from the E-5 server, the E5 Client Configuration adapter sends the business process an error.
Restrictions	There can be multiple configurations of this service.
Testing considerations	See <i>Testing the E5 Client Configuration Adapter</i> on page 358.

The following table provides an overview of the E5 Server Configuration adapter:

System name	None
Graphical Process Modeler (GPM) categories	All Services, Internet B2B
Description	The E5 Server Configuration adapter is a set of model business processes that acts as a server by receiving and storing documents from an E-5 client. These documents can also be retrieved by E-5 clients.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	E5 Server Configuration service, E5 Obtain service
Application requirements	For information on how to deploy the E5 Server Configuration adapter for Application, see the User Manual.
Business process context considerations	No
Returned status values	When applicable error messages are returned to the E-5 client.
Restrictions	There may be multiple configurations of this service.
Testing considerations	See <i>Testing the E-5 Server Configuration Adapter</i> on page 359.

Requirements

To implement, configure, and use the E-5 2000 adapter, you should be familiar with:

- The E-5 2000 specification (E-5 version 2.0 or AIAG Guideline for Electronic Commerce Message Routing).

- XML concepts

- How process data and documents are handled in Application

- BPML concepts

- Trading partner setup

- Perimeter server configuration

For the E-5 2000 adapters to work correctly, verify that you have installed and configured a Perimeter server to work with the HTTP Server adapter.

How the E-5 2000 Adapter Works

The E5 Client Configuration adapter is a set of business processes that can be used within Application to facilitate communication with an E-5 server. The E5 Client Configuration adapter uses Application trading profiles and the HTTP Send adapter in conjunction with the custom developed E5 Client Configuration service and other internal Application services to send documents and retrieve them from an outside E-5 server. The E-5 business processes are customizable examples of how a customer can use Application services to communicate with an E-5 server.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Application and will be replaced with the HTTP Client adapter and its related services. See *Retiring and Removed Services and Adapters*.

The Application E5 Client Configuration service enables you to specify parameters that are required to send E-5 requests. See *Configuring the E5 Client Configuration and E5 Server Configuration Services* on page 351 for more detailed information.

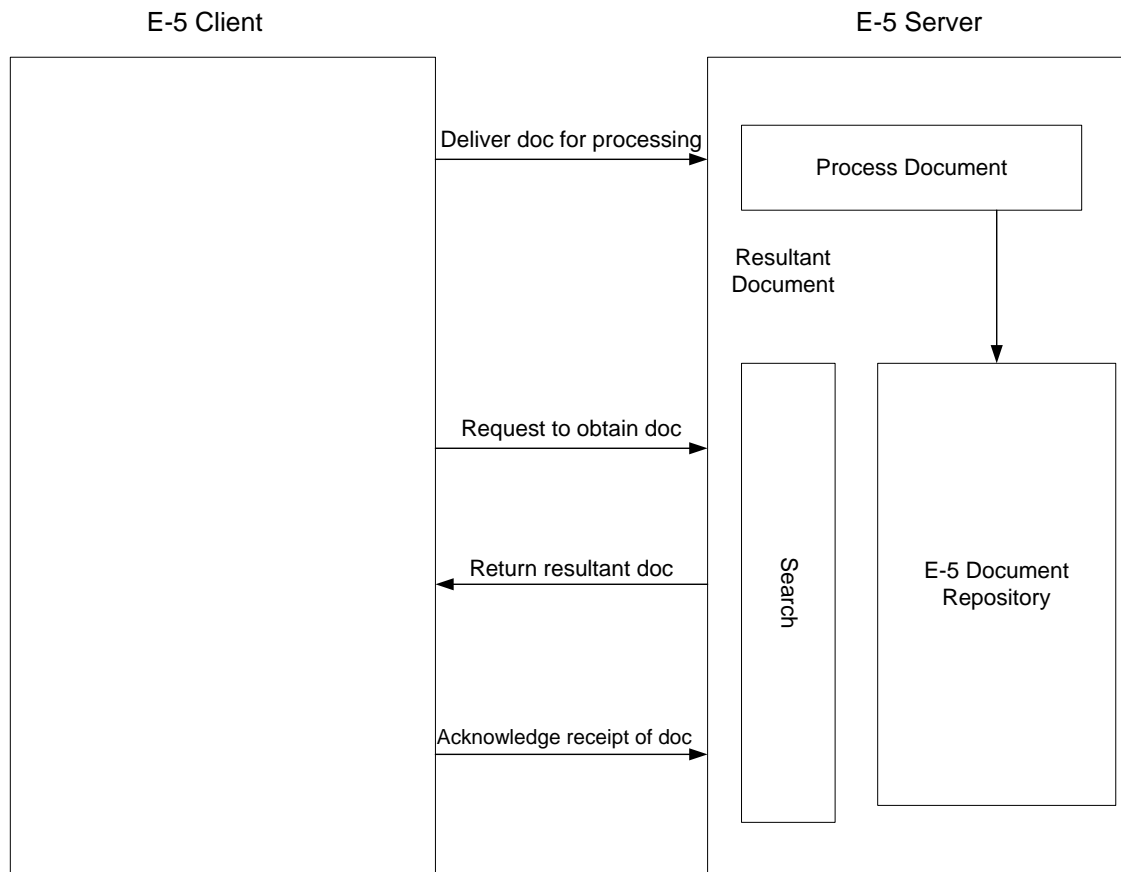
The E5 Server Configuration adapter is a set of business processes used within Application to facilitate communication with an E-5 client. The E5 Server Configuration adapter uses the Application HTTP Server adapter in conjunction with the custom-developed E5 Server Configuration service, E5 Obtain service, and other internal Application services to receive and store documents as well as retrieve documents requested by E-5 clients. The E-5 business processes are customizable examples of how a customer can use Application services to communicate with an E-5 client.

The E5 Server Configuration service enables you to specify parameters that are required to respond to E-5 requests. See *E5 Server Adapter Configuration* on page 352 for more detail.

The adapter will receive unsolicited messages in the proper format and start a BPML to process the incoming data. The adapter will also provide, through the services of the Correlation service and API, a repository of documents that can be searched by an E-5 client to obtain responses to delivered documents.

E-5 Client and Server Interactions

The following example shows standard client/server interactions that this adapter supports:

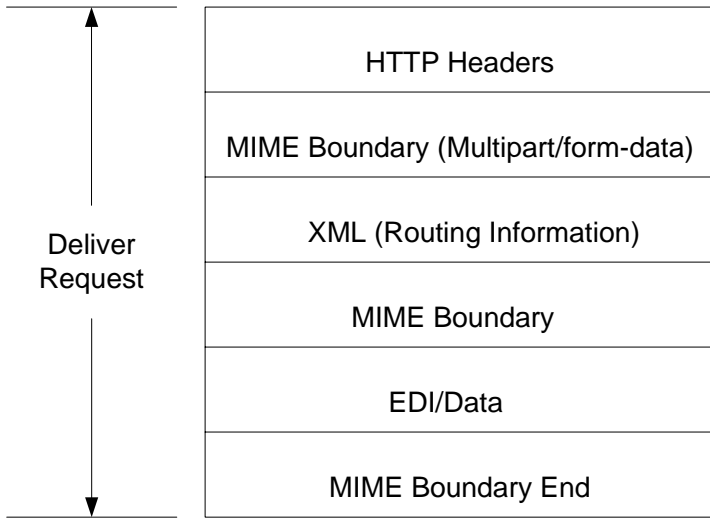


Message Formats

E-5 communication is a synchronous request/response model. Each API command has a DTD specification for the request and the response metadata that accompanies the actual data in the transfer. The response should be received in the same HTTP session.

The following figure shows a typical message format with metadata in the form of XML and the data being transferred. All messages are packaged in MIME formatting. The E-5 2000 adapter uses the HTTP Server adapter and the HTTP Send adapter or the new HTTP Client adapter. The transport mechanism to

communicate with the far-end E-5 server is HTTP or HTTP/S. Communications are secure according to those standards.



Note: Follows RFC 1867 (Form-based File Upload in HTML)

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Application and will be replaced with the HTTP Client adapter and its related services. See *Retiring and Removed Services and Adapters*

Implementing the E-5 2000 Adapter

To implement the E-5 2000 adapter, complete the following tasks:

1. Activate your license for the E5 Client Configuration and E5 Server Configuration adapters. See *An Overview of Implementing Services*.
2. Configure trading profiles for E-5. The following sample trading partner configurations are located in the `install_dir\e5 samples\TradPartnerProfiles` folder:
 - ◆ E5TPP.xml – non-SSL trading partner configuration
 - ◆ E5SSLTPP.xml – SSL trading partner configuration

Special configuration of the server transport is required if using SSL:

- ◆ Specify port 443 in URL. You might need to leave the URL as HTTP instead of HTTPS, to avoid potential issues sometimes experienced when using JAVA to decode HTTPS as a URL.
- ◆ On the Security page, select Must for SSL, STRONG for Cipher strength, and add the certificate you use

3. Configure the DTDs using the Application Map Editor:
 - a. Load the DTDs from the `si_install\e5 samples\dtds` folder.
 - b. Map the DTD corresponding to the desired request to your information.

Special instructions for mapping the Obtain Submit DTD:

- ◆ Go to withs->between->Attributes->name. Right click on name and select properties. Select the Type tab and in the default selection click the button for Implied, the default value should now be grayed out.
- ◆ Go to withs->equals->Attributes->name. Right click on name and select properties. Select the Type tab. Select CDATA for the Attribute Type. In the default selection click the button for Implied, the default value should now be grayed out.
- ◆ Go to bys->by->Attributes->name. Right click on name and select properties. Select the Type tab. Select CDATA for the Attribute Type. In the default selection click the button for Implied, the default value should now be grayed out.

The E-5 2000 adapter is dependent on the DTDs that are packaged with the adapter. Changes to the DTDs loaded into Application may cause adapter failure.

4. Create configurations for the E5 Client Configuration and E5 Server Configuration services. See *Creating a Service Configuration*.
5. Configure the E5 Client Configuration and E5 Server Configuration services. See *Configuring the E5 Client Configuration and E5 Server Configuration Services* on page 351.
6. Edit the following preconfigured configurations:
 - ◆ E5 HTTP Send adapter

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Application and will be replaced with the HTTP Client adapter and its related services. See *Retiring and Removed Services and Adapters*.
 - ◆ E5 Lightweight JDBC adapter: Define the following configuration fields:
 - Start a new business process - select This Lightweight JDBC adapter will not start a new business process
 - Pool Name - select the pool name
7. Use the E-5 adapter in a business process.

Configuring the E5 Client Configuration and E5 Server Configuration Services

To configure the E5 Client Configuration and E5 Server Configuration services, you must specify field settings in Application.

E5 Client Adapter Configuration

To configure the E5 Client adapter:

1. In Application, edit the following fields, as appropriate:

The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Application and will be replaced with the HTTP Client adapter and its related services. See *Retiring and Removed Services and Adapters*.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
E5 Sending Contract Name (E5_CONTRACTID)	Choose the trading profile contract name for sending to the remote E5 server. Required.
E5 Remote Username (E5_USERNAME)	Remote E5 server username. Required. This username was created as part of the E5 HTTP Send adapter configuration.
E5 Authentication (E5_AUTHENTICATION)	To use E5 server authentication, type TRUE; otherwise, type FALSE. Required.
Certificate Name(s) (E5_CERTIFICATENAME)	If SSL is being used, enter the certificate name(s). Required if using SSL.

2. Exchange certificates with the E-5 server if using SSL. Check in the E-5 certificates as a trusted certificate. If Application uses a CA certificate, check in the E-5 certificate as a CA certificate.

E5 Server Adapter Configuration

To configure the E5 Server Configuration adapter:

1. In Application, edit the following fields, as appropriate:

Note: It is not recommended that you create more than one instance of this service.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
E5 DTD Directory	Path to where the E5 DTDs are stored. Required.
Local E5 API URI	URI where an outside E5 server may send an E5 API Request to Application. Required.
Local URI for the E5 API Result DTD	URI where an outside E5 server may access the E5 API Result DTD on Application. Required.
Local E5 Deliver URI	URI where an outside E5 server may send an E5 Deliver Request to Application. Required.
Local URI for the E5 Deliver Submit DTD	URI where an outside E5 server may access the E5 Deliver Submit DTD on Application. Required.
Local URI for the E5 Deliver Result DTD	URI where an outside E5 server may access the E5 Deliver Result DTD on Application. Required.
Local E5 Obtain URI	URI where an outside E5 server may send an E5 Obtain Request to Application. Required.
Local URI for the E5 Obtain Submit DTD	URI where an outside E5 server may access the E5 Obtain Submit DTD on Application. Required.
Local URI for the E5 Obtain Result DTD	URI where an outside E5 server may access the E5 Obtain Result DTD on Application. Required.
Local E5 Acknowledge URI	URI where an outside E5 server may send an E5 Acknowledge Request to Application. Required.
Local URI for the E5 Acknowledge Submit URI DTD	URI where an outside E5 server may access the E5 Acknowledge Submit DTD on Application. Required.
Local URI for the E5 Acknowledge Result URI DTD	URI where an outside E5 server may access the E5 Acknowledge Result DTD on Application. Required.
Local E5 Deliver Loop Back Test URI	URI where an outside E5 server may send an E5 Deliver Loop Back Test Request to Application. Required.
E5 Template Path	Path to the E5 server templates. Required.

2. Install and configure a Perimeter server for use with the E-5 adapter.
3. Create a configuration of the HTTP Server adapter for use with the E-5 adapter. See *Creating a Service Configuration* and *HTTP Server Adapter*, using the following E-5 specific settings for the fields indicated:

- ◆ User Authentication field: Yes

You must create the user on the server side, then give the username and password to the E-5 client.

- ◆ URIs field: The following table shows a sample configuration of the URIs for the HTTP Server adapter to support the E-5 server business processes. Set up your HTTP Server adapter in a similar fashion before communicating with the far end E-5 client.

URI	BPML To Run	Raw Messages
/b2bhttp/inbound/E5Server	E5Server_Receive_All	Yes
/b2bhttp/inbound/E5ServerAPI	E5Server_Receive_APIRequest	No
/b2bhttp/inbound/E5ServerLoopBackTest	E5Server_LoopBackTest	Yes
/b2bhttp/E5_V20_APIs_Result.dtd	E5Server_DTD_Response	Yes
/b2bhttp/E5_V20_Deliver_Submit.dtd	E5Server_DTD_Response	Yes
/b2bhttp/E5_V20_Deliver_Result.dtd	E5Server_DTD_Response	Yes
/b2bhttp/E5_V20_Obtain_Submit.dtd	E5Server_DTD_Response	Yes
/b2bhttp/E5_V20_Obtain_Result.dtd	E5Server_DTD_Response	Yes
/b2bhttp/E5_V20_Acknowledge_Submit.dtd	E5Server_DTD_Response	Yes
/b2bhttp/E5_V20_Acknowledge_Result.dtd	E5Server_DTD_Response	Yes
/b2bhttp/inbound/E5Server_Obtain	E5Server_Receive_Obtain	Yes

E5 Client Configuration Adapter Business Processes

The following table describes the predefined business processes associated with the E5 Client Configuration adapter:

BPML Name	Usage Description
E5Client_Acknowledge_Submit.bpml	<ul style="list-style-type: none"> ◆ Run by E5Client_StartSubmit. ◆ Runs E5_MIME_Encode. ◆ Tells the E-5 server to mark the document corresponding to the Transaction ID as acknowledged.
E5Client_Deliver_Submit.bpml	<ul style="list-style-type: none"> ◆ Run by E5Client_StartSubmit. ◆ Runs E5_MIME_Encode. ◆ Sends meta data and a Document to the E-5 server.

BPML Name	Usage Description
E5Client_GetAPI.bpml	<ul style="list-style-type: none"> ◆ Run by E5Client_StartSubmit. ◆ Runs E5Client_Release_B2B. ◆ Requests the API Definition from the E-5 server.
E5Client_LoopBackTest.bpml	<ul style="list-style-type: none"> ◆ Run from business user when performing the E5 Loop Back Test. ◆ Runs E5_MIME_Encode. ◆ Sends meta data and a Document to the E-5 server as a communication/connection test.
E5Client_Obtain_Submit.bpml	<ul style="list-style-type: none"> ◆ Run by E5Client_StartSubmit. ◆ Runs E5_MIME_Encode, E5_MIME_Decode_Obtain. ◆ Requests Document(s) or a list of Document(s) from the E-5 server.
E5Client_Release_B2B.bpml	<ul style="list-style-type: none"> ◆ Run by E5Client_GetAPI. ◆ Releases the B2B parameters in ProcessData.
E5Client_StartSubmit.bpml	<ul style="list-style-type: none"> ◆ Run from business user when performing the E5 API Definition, Deliver Submit, Obtain Submit, and Acknowledge Submit functions. ◆ Runs E5Client_GetAPI, E5Client_Deliver_Submit, E5Client_Obtain_Submit, E5Client_Acknowledge_Submit.
E5_MIME_Decode_Obtain.bpml	<ul style="list-style-type: none"> ◆ Run by E5Client_Obtain_Submit, E5Client_LoopBackTest. ◆ Decodes multiple MIME documents.
E5_MIME_Encode.bpml	<ul style="list-style-type: none"> ◆ Run by E5Client_Deliver_Submit, E5Client_Obtain_Submit, E5Client_Acknowledge_Submit, E5Client_LoopBackTest. ◆ Encodes MIME documents.

Facade BPML

Use a facade BPML to run the predefined business processes listed in *E5 Client Configuration Adapter Business Processes* on page 354. The facade contains the environmental setup information that enables the business process to communicate successfully with E-5 servers. You should customize the facade to meet the specific needs of your company.

BPML	Description
API Definition	<ul style="list-style-type: none"> ◆ Run E5ClientConfig. ◆ Place no documents in ProcessData. ◆ Run E5Client_StartSubmit.bpml.

BPML	Description
Deliver Submit	<ul style="list-style-type: none"> ◆ Run E5ClientConfig. ◆ Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData. ◆ Run E5Client_StartSubmit.bpml.
Obtain Submit	<ul style="list-style-type: none"> ◆ Run E5ClientConfig. ◆ Place the Obtain Submit document as PrimaryDocument in ProcessData. ◆ Run E5Client_StartSubmit.bpml.
Acknowledge Submit	<ul style="list-style-type: none"> ◆ Run E5ClientConfig. ◆ The Acknowledge Submit document should be placed as PrimaryDocument in ProcessData. ◆ Run E5Client_StartSubmit.bpml.
Loop Back Test	<ul style="list-style-type: none"> ◆ Run E5ClientConfig. ◆ Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData. ◆ Run E5Client_LoopBackTest.bpml.

E5 Server Configuration Adapter Business Processes

The following table describes the predefined business processes associated with the E5 Server Configuration adapter:

Note: Business processes (instances) using this model should remain in the system for the specified amount of time to 30 days for some of the business processes, so that the documents will stay in the repository for 30 days.

BPML Name	Usage Description
E5Server_Acknowledge.bpml	<ul style="list-style-type: none"> ◆ Run by E5Server_Receive_All. ◆ Runs E5Server_Input_ErrorMsg. ◆ Records that a document has been acknowledged. ◆ Recommended to change system time to 30 days.
E5Server_DTD_Response.bpml	<ul style="list-style-type: none"> ◆ Run by HTTP Server adapter. ◆ Provides DTDs to clients through a URI.
E5Server_Deliver.bpml	<ul style="list-style-type: none"> ◆ Run by E5Server_Receive_All. ◆ Invokes E5Server_Input_ErrorMsg, E5Server_Deliver_ProcessDocument, E5Server_StoreValues. ◆ Stores meta data and Documents in the Correlation service.

BPML Name	Usage Description
E5Server_Deliver_ProcessAttachment.bpm1	<ul style="list-style-type: none"> ◆ Run by E5Server_Deliver_ProcessAttachment. ◆ Obtains the attachment and sets the obtain date, acknowledge date, and acknowledge flag. Business user can then process the attachment document ◆ Recommended to change system time to 30 days.
E5Server_Deliver_ProcessDocument.bpm1	<ul style="list-style-type: none"> ◆ Run by E5Server_Deliver. ◆ Invokes E5Server_Deliver_ProcessAttachment.bpm1 ◆ If the Business User wants to process the attachment once it is delivered, the Business User will have to modify the rule in this BP.
E5Server_Input_ErrorMsg.bpm1	<ul style="list-style-type: none"> ◆ Run by E5Server_Acknowledge, E5Server_Deliver, E5Server_Obtain, E5Server_Receive_APIRequest. ◆ Inputs Error Messages into the results templates.
E5Server_LoopBackTest.bpm1	<ul style="list-style-type: none"> ◆ Run by HTTP Server adapter. ◆ Invokes E5_MIME_Decode, E5_MIME_Encode_Obtain, E5Server_Input_ErrorMsg. ◆ Returns the documents received to the E5 client requester.
E5Server_Obtain.bpm1	<ul style="list-style-type: none"> ◆ Run by HTTP Server adapter. ◆ Invokes E5Server_Input_ErrorMsg, E5_MIME_Encode_Obtain. ◆ Retrieves documents by request. ◆ Recommended to change system time to 30 days.
E5Server_Receive_APIRequest.bpm1	<ul style="list-style-type: none"> ◆ Run by HTTP Server adapter. ◆ Invokes E5Server_Input_ErrorMsg. ◆ Returns an API Definition document.
E5Server_Receive_All.bpm1	<ul style="list-style-type: none"> ◆ Run by HTTP Server adapter. ◆ Invokes E5_MIME_Decode, E5Server_Acknowledge, E5Server_Deliver, E5Server_Input_ErrorMsg. ◆ Receives E5 client requests and sorts them according to type.
E5Server_StoreValues.bpm1	<ul style="list-style-type: none"> ◆ Run by E5Server_Deliver. ◆ Stores documents with name/value pairs in the Correlation service. ◆ Recommended to change system time to 30 days.
E5_MIME_Decode.bpm1	<ul style="list-style-type: none"> ◆ Run by E5Server_Receive_All, E5Server_Receive_Obtain, E5Server_LoopBackTest. ◆ Decodes MIME documents.
E5_MIME_Encode_Obtain.bpm1	<ul style="list-style-type: none"> ◆ Run by E5Server_Obtain, E5Server_LoopBackTest. ◆ Encodes multiple MIME documents.

BPML Name	Usage Description
E5Server_Receive_Obtain.bpml	<ul style="list-style-type: none"> ◆ Run by HTTP Server adapter. ◆ Invokes E5Server_Obtain.

BPML Error Handling

There are two categories of adapter errors:

Service failures – Can be detected by on fault BPML constructs.

Detectable errors – Must be checked for specifically.

Functionality Supported for Application

The following list contains the subset of Obtain features supported by this adapter:

Support up to four “equals” search parameters

Wildcard character searches are supported

No support for “between” searching

No support for “by” searching

Acknowledgement of Document(s) removes it from the list of possible documents to be returned in an Obtain search. This adapter follows the standard for not Obtaining Documents that have an Acknowledge DateTime set, due to contradictions in the E-5 specification.

Testing the E5 Client Configuration Adapter

To test the E5 Client Configuration adapter:

1. Configure the E5ClientConfig service instance.
2. Using the following table, perform the steps for each E-5 operation, and then verify the results:

E-5 Operation	Steps	Result
API Definition	<ol style="list-style-type: none"> 1 Invoke E5ClientConfig. 2 Place no documents in ProcessData. 3 Invoke E5Client_StartSubmit.bpml. 	API Definition Results document should be the PrimaryDocument.

E-5 Operation	Steps	Result
Deliver Submit	<ol style="list-style-type: none"> 1 Invoke E5ClientConfig. 2 Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData. 3 Invoke E5Client_StartSubmit.bpml. 	Deliver Results document should be the PrimaryDocument.
Obtain Submit	<ol style="list-style-type: none"> 1 Invoke E5ClientConfig. 2 Place the Obtain Submit document as PrimaryDocument in ProcessData. 3 Invoke E5Client_StartSubmit.bpml. 	Obtain Results documents and attachment documents, named RoutingDocDecoded_1, AttachmentDocDecoded_1, RoutingDocDecoded_2, etc.
Acknowledge Submit	<ol style="list-style-type: none"> 1 Invoke E5ClientConfig. 2 Place the Acknowledge Submit document as PrimaryDocument in ProcessData. 3 Invoke E5Client_StartSubmit.bpml. 	Acknowledge results document should be the PrimaryDocument.
Loop Back Test	<ol style="list-style-type: none"> 1 Invoke E5ClientConfig. 2 Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData. 3 Invoke E5Client_LoopBackTest.bpml. 	RoutingDocDecoded_1 and AttachmentDocDecoded_1

Testing the E-5 Server Configuration Adapter

To test the E-5 Server Configuration adapter:

1. Configure service instances for the E-5 Server Configuration service, E-5 Obtain service, and HTTP Server adapter.

2. Using the following table as a guide, test each of the E-5 operations by verifying that the E-5 client submitted the correct document:

E-5 Operation	Description
API Definition	<ol style="list-style-type: none"> 1 The client sends the request (configured per the specifications in the AIAG Guideline for Electronic Commerce Message Routing) to the URI that has been set up to handle API requests. This request is received through the HTTP Server adapter 2 Invokes the E5Server_Receive_APIRequest.bpml. 3 An API definition document is created using a template placed in a predefined location and the document is returned to the E-5 client.
Deliver Submit	<ol style="list-style-type: none"> 1 The client sends the meta data and document according to the specifications in the AIAG Guideline for Electronic Commerce Message Routing. The request is sent to the URI that has been set up to handle general E-5 server requests and is received through the HTTP Server adapter 2 Invokes the E5Server_Receive_All.bpml. 3 The document is stored in the Correlation service along with name/value pairs based on the meta data and other parameters. 4 A Deliver results document is created using a template placed in a pre-defined location and the document is returned to the E-5 client.
Obtain Submit	<ol style="list-style-type: none"> 1 The client sends the request according to the specifications in the AIAG Guideline for Electronic Commerce Message Routing. Note: Bys and Betweens are not supported in this E5 server. 2 The request is sent to the URI that has been set up to handle general E-5 server requests and is received through the HTTP Server adapter. 3 Invokes the E5Server_Receive_All.bpml. The E5 Obtain service is used to search the Correlation service for the requested information. 4 The Obtain results are returned to the E-5 client.
Acknowledge Submit	<ol style="list-style-type: none"> 1 The client sends the request according to the specifications in the AIAG Guideline for Electronic Commerce Message Routing. The request is sent to the URI that has been set up to handle general E-5 server requests and is received through the HTTP Server adapter. 2 Invokes the E5Server_Receive_All.bpml. The acknowledgeFlag and acknowledgeDateTime are updated in the Correlation service. 3 The acknowledge results are returned to the E-5 client.

ebXML BPSS Correlation Service

The following table provides an overview of the ebXML BPSS Correlation service:

System name	BPSSCorrelation
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Interfaces with the Correlation API to track transactions, activities and documents going out and coming in as part of an ebXML message.
Business usage	Tracks the: <ul style="list-style-type: none">◆ BPSS transaction◆ BPSS activities within a transaction◆ Documents within a BPSS activity
Usage example	A business process that needs to track the BPSS activities can invoke this service by passing the required parameters.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	None
Restrictions	None

Implementing the ebXML BPSS Correlation Service

To implement the ebXML BPSS Correlation service, complete the following tasks:

1. Create a ebXML BPSS Correlation service configuration. See *Managing Services and Adapters*.
2. Configure the ebXML BPSS Correlation service. For information, see *Configuring the ebXML BPSS Correlation Service*.
3. Use the ebXML BPSS Correlation service in a business process.

Configuring the ebXML BPSS Correlation Service

To configure the ebXML BPSS Correlation service, you must define following fields in the GPM:

Field	Description
Config	Name of the service configuration.
conversation_id	The conversation_id of the ebXML message. Required.
cpa_id	A valid Collaboration Protocol Agreement (CPA) ID. Required.
action	Indicates to the service what action needs to be performed. Required. Valid values are: <ul style="list-style-type: none">◆ SELECT◆ INSERT◆ UPDATE
DBValues	The amount of information passed within the DBValues node varies depending on the action. Required.

ebXML BPSS Correlation Service Business Process Usage

When the action is SELECT, an attempt is made to retrieve the transaction ID. The status values shown in the following examples are:

0 = WAITING

1 = COMPLETED

9 = FAILED

The InDoc should contain the following values when the action is SELECT:

```
<inDoc>
  <DBValues>
    <tp_name>Sterling 2</tp_name>
    <status>0</status>
  </DBValues>
</inDoc>
```

When the action is INSERT, an attempt is made to insert the values shown in the following example:

```
<inDoc>
  <DBValues>
    <tp_name>Sterling 2</tp_name>
    <status>0</status>
    <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
    <time_to_perform>1440</time_to_perform>
    <role_name>buyer</role_name>
    <start_time>1062126842123</start_time>
    <transaction_type>IssuePurchaseOrder</transaction_type>
  </DBValues>
</inDoc>
```

When the action is UPDATE, an attempt is made to insert the values shown in the following example. The status values shown are:

1 = COMPLETED

9 = FAILED

```
<inDoc>
  <DBValues>
    <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
    <activity>RequestPurchaseOrder</activity>
    <doc_id>sgcentennial:4bdb20:f74ba603b7:-6166</doc_id>
    <ctr>1</ctr>
    <workflow_id>7015</workflow_id>
    <status>1</status>
    <end_time>1062140580230</end_time>
    <exception_workflow_id>0</exception_workflow_id>
  </DBValues>
</inDoc>
```

Output from Service to Business Process

The following parameters are passed from the service to a business process:

Field	Description
OutValues	Contains the retrieved values. Only applicable when the input action parameter is SELECT. Optional.

The following output values will only be returned if a successful selection is made:

```
<OutValues>
  <OutDoc>
    <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
    <time_to_perform>2880</time_to_perform>
  </OutDoc>
</OutValues>
```

The following parameters need to be passed to the service when it is being Run with the output message set to processActivity:

Field	Description
action	Indicates to the service what action needs to be performed. Required. Valid values are: <ul style="list-style-type: none">◆ SELECT◆ INSERT◆ UPDATE

When the action is SELECT, an attempt is made to retrieve the count of the activity name corresponding to the transaction ID:

```
<inDoc>
  <DBValues>
    <trx_id>sgcentennial:4bdb20:f74ba603b7:-71f9</trx_id>
    <activity>RequestPurchaseOrder</activity>
  </DBValues>
</inDoc>
```

When the action is SELECT_ID, an attempt is made to retrieve the business process ID of the activity:

```
<inDoc>
  <DBValues>
    <trx_id>sgcentennial:4bdb20:f74ba603b7:-71f9</trx_id>
    <activity>RequestPurchaseOrder</activity>
  </DBValues>
</inDoc>
```

When the action is INSERT, an attempt is made to insert the values in the following example:

```
<inDoc>
  <DBValues>
    <trx_id>sgcentennial:4bdb20:f74ba603b7:-71f9</trx_id>
    <time_to_perform>2880</time_to_perform>
    <workflow_id>7015</workflow_id>
    <timeStamp>1062126848411</timeStamp>
    <activity>RequestPurchaseOrder</activity>
    <process_flow>send</process_flow>
  </DBValues>
</inDoc>
```

Output from Service to Business Process

The following parameters are passed from the service to a business process when the input action is SELECT or SELECT_ID:

Field	Description
OutValues	Contains the retrieved values. Only applicable when the input action parameter is set to SELECT or SELECT_ID.

The following output values will only be returned if a successful selection (that is, action=SELECT) is made:

```
<OutValues>
  <OutDoc>
    <result>1</result>
  </OutDoc>
</OutValues>
```

The following output values will only be returned if a successful workflow_id selection (that is, action=SELECT_ID) is made:

```
<OutValues>
  <OutDoc>
    <workflow_id>1018</ workflow_id >
  </OutDoc>
</OutValues>
```

The following parameters need to be passed from the business process to the service when it runs with the output message set to processDocument:

Field	Description
action	Indicates to the service what action needs to be performed. Required. Valid value is INSERT.

When the action is INSERT, an attempt is made to insert the values shown below. The ctr value show below indicates the nth document. Typically, the insertion of the documents is done in a loop:

```
<inDoc>
  <DBValues>
    <trx_id>sgcentennial:4bdb20:f74ba603b7:-71f9</trx_id>
    <activity>ConfirmPurchaseOrder</activity>
    <doc_id>sgcentennial:4bdb20:f74ba603b7:-6166</doc_id>
    <ctr>1</ctr>
  </DBValues>
</inDoc>
```

Business Process Example

The following example assumes that the nodes DBValues are found in the process data:

```
<operation>
  <participant name="BPSSCorrelation"/>
  <output message="processTransaction">
    <assign to="conversation_id" from="conversation_id/text()"/>
    <assign to="cpa_id" from="cpa_id/text()"/>
    <assign to="action" from="'SELECT'"/>
    <assign to="inDoc" from="DBValues"/>
  </output>
  <input message="TransactionResponse">
    <assign to="DBValues/trx_id" from="OutValues/OutDoc/trx_id/text()"/>
    <assign to="DBValues/time_to_perform"
from="OutValues/OutDoc/time_to_perform/text()"/>
  </input>
</operation>

<operation>
  <participant name="BPSSCorrelation"/>
  <output message="processActivity">
    <assign to="action" from="'SELECT'"/>
    <assign to="inDoc" from="DBValues"/>
  </output>
```

```
<input message="ActivityResponse">
  <assign to="DBValues/result" from="OutValues/OutDoc/result/text()"/>
</input>
</operation>

<operation>
  <participant name="BPSSCorrelation"/>
  <output message="processDocument">
    <assign to="action" from="'INSERT'"/>
    <assign to="inDoc" from="DBValues"/>
  </output>
  <input message="DocumentResponse">
    <assign to="." from="*" />
  </input>
</operation>
```

ebXML BPSS Doc Parsing Service

The following table provides an overview of the ebXML BPSS Doc Parsing service:

System name	BPSSDocParser
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Validates the document based on the Conditional Expression specified in the BPSS.
Business usage	It helps to determine the SUCCESS or FAILURE of the state of the transaction.
Usage example	A business process that needs to determine the end state of the transaction can invoke this service by passing the required parameters.
Preconfigured?	Must be installed and deployed before it can run. There are no configuration parameters required.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Not applicable
Business process context considerations	No
Returned status values	None
Restrictions	No

Implementing the ebXML BPSS Doc Parsing Service

To implement the ebXML BPSS Doc Parsing service, complete the following tasks:

1. Create an ebXML BPSS Doc Parsing service configuration. See *Managing Services and Adapters*.
2. Configure the ebXML BPSS Doc Parsing service. For information, see *Configuring the ebXML BPSS Doc Parsing Service*.
3. Use the ebXML BPSS Doc Parsing service in a business process.

Configuring the ebXML BPSS Doc Parsing Service

To configure the ebXML BPSS Doc Parsing service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
documentId	The ID of the document. Required.
documentKey	Contains the value PrimaryDocument.
expression	The conditional expression as specified in the BPSS. Enclose the expression with ' instead of single quotation marks. Required.

Output from Service to Business Process

The following values are passed from the service to a business process:

Field	Description
validityStatement/validate/value	Indicates if the documents is validated correctly. Possible values are true and false.

The following examples show an input message coming back to the business process:

Example 1

```
<validityStatement>
  <validate>
    <value>true</value>
  </validate>
</validityStatement>
```

Example 2

```
<validityStatement>
  <validate>
    <value>true</value>
  </validate>
</validityStatement>
```

Business Process Example

This business process assumes that the output message values are found in the process data:

```
<operation name="ValidateDoc">
  <participant name="BPSSDocParser"/>
  <output message="validateCondExpr">
    <assign to="documentId" from="documents/document[1]/doc:document-id/text()"/>
```



```
<assign to="expression"
from="string(ProcessSpecification/Package/BinaryCollaboration[@name=//binaryCollaboration/text()]/Success[@fromBusinessState=//businessTransactionActivity/text()]/ConditionExpression/@expression)"/>
</output>
<input message="validateCondResponse">
<assign to="result" from="validityStatement/validate/value/text()"/>
</input>
</operation>
```

ebXML Business Service Handler (BSI) service

The following table provides an overview of the ebXML Business Service Handler (BSI) service:

System name	ebXMLBSIService
Graphical Process Modeler (GPM) categories	All Services, Internet B2B - ebXML
Description	Invokes the Business Service Handler to interpret and execute activities defined in the BPSS. Note: BPSS 1.05 is supported.
Business usage	<ul style="list-style-type: none">◆ Executes either inbound or outbound processing for BPSS.◆ Determines start, finish and transition activities.◆ Invokes appropriate services to do document schema validation.
Usage example	A business process that needs to execute activities specified in the BPSS.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Not applicable
Business process context considerations	No
Returned status values	None
Restrictions	No
Persistence level	System default
Testing considerations	None

Output from Service to Business Process

The following table describes the output from the ebXML BSI service to the business process:

Parameter	Description
wfc	Contents of the updated business process context. Required.

Output from Business Process to Service

The following table describes the output from the business process to the ebXML BSI service:

Parameter	Description
wfc	The contents of the business process context. Required.

Business Process Example

The following example assumes that the relevant activity_state (if needed) is set.

```
<operation>
  <participant name="ebXMLBSIService"/>
  <output message="XOut">
    <assign to="." from="*" />
  </output>
  <input message="XIn">
    <assign to="." from="*" />
  </input>
</operation>
```

Output Parameters

This section contains information about calling the ebXML service for outbound and inbound activities.

Output Parameters – Outbound Processing

In processing outbound activities, there are two possible ways of calling the ebXML BSI service. The first way, with the activity_flag set to pre indicates preprocessing of activities before the internal process runs. The second way, with the activity_flag set to post, indicates postprocessing of activities after the internal process runs.

```
<assign to="activity_state" from="'pre'"/>
<operation name="One">
  <participant name="ebXMLBSIService"/>
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

```
<assign to="activity_state" from="'post'"/>
<operation name="One">
  <participant name="ebXMLBSIService"/>
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
```

```
        <assign to="." from="*" />
    </input>
</operation>
```

Output Parameters – Inbound Processing

When processing inbound activities, just call the service, as shown in the following example:

```
<operation name="One">
  <participant name="ebXMLBSIService" />
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

Implementing the ebXML BSI Service

To implement the ebXML BSI service for use in a business process:

1. Create an ebXML BSI service configuration. See *Managing Services and Adapters*.
2. Use the ebXML BSI service in a business process.

ebXML CPA Lookup Service

The following table provides an overview of the ebXML CPA Lookup service:

System name	ebXMLCPALookupService
Graphical Process Modeler (GPM) categories	All Services, Internet B2B - ebXML
Description	Retrieve CPA information based on service/action or business request. Note: CPA 2.0 is supported.
Business usage	Used for retrieving CPA information before creating a new ebXML outbound message or before validating an ebXML inbound message.
Usage example	A business process that needs to retrieve specific CPA information can invoke this service by passing the required parameters.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	None
Restrictions	No
Persistence level	System default
Testing considerations	To test a configuration of the ebXML CPA Lookup service, provide the the necessary input parameters.

Output from Service to Business Process

The following table describes the output from the ebXML CPA Lookup service to the business process:

Parameter	Description
cpa	CPA info

Output from Business Process to Service

The following table describes the output from the business process to the ebXML CPA Lookup service:

Parameter	Description
cpa_id	A unique value that identifies the alias name of CPA. Required.
service	A unique value that identifies the service of the ebXML message
serviceType	A unique value that identifies the service type of the ebXML message. Optional.
action	A unique value that identifies the action of the ebXML message. Optional.
b2b-message-mode	A flag for ebXML Lite which is passed from trading partner. Valid values are Send and Respond. Optional.
BPSSMode	A flag that identifies whether ebXML message is outbound or inbound. Valid values are Send and Receive. Required.
thisPartyId	A unique value that identifies the party name or ID of the current Trading Partner. Optional.
fromb2bInbound	A flag that identifies that the message is not the first request. Valid values are "" or true. Optional.
binaryCollaboration	A unique value that identifies the binaryCollaboration of the ebXML message. Optional.
businessTransactionActivity	A unique value that identifies the businessTransactionActivity of the ebXML message. Optional.
requestOrResponseAction	A unique value that identifies the requestOrResponseAction of the ebXML message. Optional.

Business Process Example

The following example illustrates using the ebXML CPA Lookup service in a business process:

```
<process name="testEBXMLCPALookupService">
  <sequence>
    <operation name="ebXMLCPALookup">
      <participant name="ebXMLCPALookupService"/>
      <output message="ebXMLCPALookupInputMessage">
        <assign to="." from="*" />
      </output>
      <input message="ebXMLCPALookupOutputMessage">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Implementing the ebXML CPA Lookup Service

To implement the ebXML CPA Lookup service for use in a business process:

1. Create an ebXML CPA Lookup service configuration. See *Managing Services and Adapters*.
2. Use the ebXML CPA Lookup service in a business process.

ebXML Lookup Service

The following table provides an overview of the ebXML Lookup service:

System name	ebXMLLookupService
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Retrieves the content of either CPA or BPSS schema, based on the alias name of the schema and the type of schema.
Business usage	Used for the ebXML Messaging service and the BPSS Handler service to retrieve the schema information of CPA or BPSS.
Usage example	A business process that needs to retrieve CPA or BPSS schema can invoke this service by passing the required parameters.
Preconfigured?	No. Must be installed and deployed before it can run. There are no configuration parameters required.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	None
Restrictions	No
Testing considerations	To test a configuration of the ebXML Lookup service, make sure you have the CPA or BPSS schema created under Deployment/EBXML/BPSS or CPA.

How the ebXML Lookup Service Works

The ebXML Lookup service retrieves the content of either CPA or BPSS schema, based on the alias name of schema and the type of schema.

Parameters Passed from Business Process to Service

The following parameters are passed to the service when it runs with the output message set to lookupEBXML:

Field	Description
ebxmlName	A unique value that identifies the alias name of CPA or BPSS schema.
schemaType	A value that identifies the type of schema to be retrieved. Possible values are CPA and BPSS.

Business Process Example

The following example shows how the ebXML Lookup service might be used in a business process:

Retrieve CPA schema:

```
<operation>
  <participant name="ebXMLLookupService"/>
  <output message="lookupEBXML">
    <assign to="ebxmlName" from="cpaId/text()" />
    <assign to="schemaType" from="'CPA'"/>
  </output>
  <input message="getEBXML">
    <assign to="cpa-store" from="ebxmlSchema/node()" />
  </input>
</operation>
```

Retrieve BPSS schema:

```
<operation>
<participant name="ebXMLLookupService"/>
  <output message="lookupEBXML">
    <assign to="ebxmlName" from="bpsName/text()" />
    <assign to="schemaType" from="'BPSS'"/>
  </output>
  <input message="getEBXML">
    <assign to="." from="ebxmlSchema/node()" />
  </input>
</operation>
```

Implementing the ebXML Lookup Service

To implement the ebXML Lookup service, complete the following tasks:

1. Activate your license for the ebXML Lookup service. See *Installing Application*.
2. Create a ebXML Lookup service configuration. See *Managing Services and Adapters*.
3. Configure the ebXML Lookup service. See *Configuring the ebXML Lookup Service*.

4. Use the ebXML Lookup service in a business process.

Configuring the ebXML Lookup Service

To configure the ebXML Lookup service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration.
ebXMLName	Unique value that identifies the alias name of CPA or BPSS schema.
schemaType	A value that identifies the type of schema to be retrieved. Possible values are CPA and BPSS.

ebXML Manifest and Payload Service

The following table provides an overview of the ebXML Manifest and Payload service:

System name	ManifestPayloadService
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Create eb:Manifest and payloads node before constructing ebXML outbound message.
Business usage	Used for creating eb:Manifest and payloads node based on documents/document.
Usage example	A business process that needs to create eb:Manifest and payloads with documents/document input structure can invoke this service by passing the required parameters.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	None
Restrictions	No
Persistence level	System default
Testing considerations	To test a configuration of the Manifest and Payload service, provide the documents/document structure.

Output from Service to Business Process

The following table describes the output from the ebXML Manifest and Payload service to the business process:

Parameter	Description
eb:Manifest	Manifest information prepared for SOAP-ENV:Body.
payloads	Payload information prepared for mime:message/mime:body

Output from Business Process to Service

The following table describes the output from the business process to the ebXML Manifest and Payload service:

Parameter	Description
documents	A node contains the document information. Required.

Business Process Example

The following example illustrates using the ebXML Manifest and Payload service in a business process:

```
<process name="testManifestandPayloadService">
  <sequence>
    <operation name="CreateManifestPayload">
      <participant name="ManifestPayloadService"/>
      <output message="ManifestPayloadInputMessage">
        <assign to="outboundDoc" from="documents/node()"/>
      </output>
      <input message="ManifestPayloadOutputMessage">
        <assign to="." from="Manifest/node()"/>
        <assign to="." from="Payload/node()"/>
      </input>
    </operation>
  </sequence>
</process>
```

Implementing the ebXML Manifest and Payload Service

To implement the ebXML Manifest and Payload service for use in a business process:

1. Create an ebXML Manifest and Payload service configuration. See *Managing Services and Adapters*.
2. Use the ebXML Manifest and Payload service in a business process.

ebXML Profile Service

The ebXML Profile service returns a specified business process name to be run. The service is used to reuse a business process and to prevent more than one instance of the same business process from being started. The following table provides an overview of the ebXML Profile service:

System name	ebXMLProfile
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Returns the name of a business process to be run.
Business usage	Used in conjunction with ebXML messaging. It determines the business process to be run based on the CPA ID, service and action values passed as part of the ebXML message.
Usage example	A business process that correctly validates an ebXML message needs information about what other business process needs to be run in response to the ebXML message.
Preconfigured?	Must be installed and deployed before it can run. There are no configuration parameters required.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	None
Restrictions	No
Notes	The output message is a node containing the business process to be run. <pre><callback> <messageName>DoIt</messageName> <processDefinition> <workFlowName>TestServerIn</workFlowName> </processDefinition> </callback></pre>

Parameters Passed from Business Process to Service

The following parameters need to be passed to the service when it runs with the output message set to `getCallbackRequest`.

Field	Description
CPAId	Unique value that identifies a trading partner relationship as defined in the CPA specification.
Service	Identifies a set of business transactions as defined in the CPA specification.
Action	Identifies a business transaction as defined in the CPA specification.

Business Process Example

The following example shows how the ebXML Profile service might be used in a business process:

```
<operation>
  <participant name="ebXMLProfile"/>
  <output message="getCallbackRequest">
    <assign to="CPAId"
from="InboundMIME/mime:message/mime:body/mime:message[1]/mime:body/SOAP-ENV:Envelope
/SOAP-ENV:Header/eb:MessageHeader/eb:CPAId/text()" />
    <assign to="Service"
from="InboundMIME/mime:message/mime:body/mime:message[1]/mime:body/SOAP-ENV:Envelope
/SOAP-ENV:Header/eb:MessageHeader/eb:Service/text()" />
    <assign to="Action"
from="InboundMIME/mime:message/mime:body/mime:message[1]/mime:body/SOAP-ENV:Envelope
/SOAP-ENV:Header/eb:MessageHeader/eb:Action/text()" />
  </output>
  <input message="getCallbackResponse">
    <assign to="Client" from="callback/node()" />
  </input>
</operation>
```

Implementing the ebXML Profile Service

To implement the ebXML Profile service, complete the following tasks:

1. Activate your license for the ebXML Profile service. See *Installing Application*.
2. Create an ebXML Profile service configuration. See *Managing Services and Adapters*.
3. Use the ebXML Profile service in a business process.

ebXML Request Response Service

The ebXML Request Response service generates unique conversation IDs and message IDs that are used in an ebXML conversation. The following table provides an overview of the ebXML Request Response service:

System name	ebXMLRequestResponse
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Returns the name of a business process to be run.
Business usage	Used with ebXML messaging. Creates unique values of conversation and message IDs.
Usage example	A business process that is creating an ebXML message needs to generate a dialog or message ID if the current message is part of a group of messages, which should be treated as part of the same dialog.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	None
Restrictions	None

Parameters Passed from Business Process to Service

The following parameters need to be passed to the service when it runs with the output message set to `associateRequest`.

Field	Description
PartyId	Unique value that identifies a trading partner relationship as defined in the CPA specification. Required.
ConversationId	Identifies a set of business transactions as defined in the CPA specification.
MessageId	Identifies a business transaction as defined in the CPA specification.

Field	Description
handler	Handler, that is, the ID of the ebXML Request Response service. Numeric. Valid value is a system-generated ID.

Business Process Example

The following is an example of a business process that uses the ebXML Request Response service:

```
<operation>
  <participant name="ebXMLRequestResponse" />
<output message="associateRequest">
  <assign to="PartyId" from="string(cpa/counterParty/eb:PartyId)"/>
  <assign to="handler" from="thisProcessInstance/node()"/>
</output>
<input message="associateResponse">
  <assign to="conversationId" from="ConversationId/node()"/>
  <assign to="messageId" from="MessageId/node()"/>
</input>
</operation>
```

Implementing the ebXML Request Response Service

To implement the ebXML Request Response service, complete the following tasks:

1. Activate your license for the ebXML Request Response service. See *Installing Application*.
2. Create an ebXML Request Response service configuration. See *Managing Services and Adapters*.
3. Use the ebXML Request Response service in a business process.

ebXML Validation Service

The following table provides an overview of the ebXML Validation service:

System name	ebXMLValidationService
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Validates inbound ebXML message to make sure it conforms to the ebXML structure and verify the validity of the digital signature. If there is a message level exception found, returns error list. Supported ebXML stacks are: Note: ebMS 2.0 is supported.
Business usage	Used for ebXML Messaging to validate inbound message level content.
Usage example	A business process that needs to validate ebXML inbound message can invoke this service by passing the required parameters.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	None
Restrictions	No
Persistence level	System default
Testing considerations	None

Output from Service to Business Process

The following table describes the output from the ebXML Validation service to the business process:

Parameter	Description
errors	Error List. Valid value is eb:ErrorList or eb:Error. Optional.

Output from Business Process to Service

The following table describes the output from the business process to the ebXML Validation service:

Parameter	Description
inboundMIME	Node contains the inbound ebXML message. Required.
CPA	Node contains the CPA info based on the inbound ebXML message service/action. Required.
BPSSParam	Node contains the BPSS parameters information based on the return of BPSSHandler. Optional.
InboundDocId	Node contains the document-id of the inbound ebXML message. Required.

Business Process Example

The following example illustrates using the ebXML Validation service in a business process:

```
<process name="testEBXMLValidationService">
  <sequence>
    <operation name="ebXMLValidation">
      <participant name="ebXMLValidationService"/>
      <output message="ebXMLValidationInputMessage">
        <assign to="inboundMIME" from="InboundMIME/node()" />
        <assign to="CPA" from="cpa/node()" />
        <assign to="BPSSParam" from="bpssParams/node()" />
        <assign to="InboundDocId"
from="string(inbound-mime-document/doc:document-id)" />
      </output>
      <input message="ebXMLValidationOutputMessage">
        <assign to="." from="errors"/>
      </input>
    </operation>
  </sequence>
</process>
```

Implementing the ebXML Validation Service

To implement the ebXML Validation service for use in a business process:

1. Create an ebXML Validation service configuration. See *Managing Services and Adapters*.
2. Use the ebXML Validation service in a business process.

ebXML XML Digital Signature Service

The ebXML XML Digital Signature service composes and verifies digital signatures. The following table provides an overview of the ebXML XML Digital Signature service:

System name	ebXMLDSig
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > ebXML
Description	Composes and verifies digital signatures.
Business usage	<ul style="list-style-type: none">◆ Composes a digital signature (XMLDSIG) on a given node or primary document by using a private key of a specified and returns a hash value together with the original input node.◆ Validates the hash value of the input node or Primary Document by making use of a public key, either through a specified certificate or from the KeyInfo element of the Signature and indicates this with a TRUE or FALSE.
Usage example	A business process that needs a node to be digitally signed or verified can invoke this service by passing the required parameters.
Preconfigured?	Must be installed and deployed before it can run. There are no configuration parameters required.
Requires third party files?	xss4j.jar (included in Application installation)
Platform availability	All supported Application platforms.
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	None
Restrictions	None
Testing considerations	Use the correct certificates for the signing. The most frequent problem encountered is the certificates used for the signing were not created with a storepass value of <i>integrator</i> and a keypass value of <i>integrator</i> . If you receive an error with this condition, see your system administrator.

Notes	<p>Output Parameters: signRequest</p> <p>A node or updated Primary Document containing the original contents and the Signature element after the digital signature is applied.</p> <p>Output Parameters: verifyRequest</p> <p>A node containing the validity of the signature. The two possible values are:</p> <pre><validity> <valid>true</value> </validity></pre> <p>and</p> <pre><validity> <valid>>false</value> </validity></pre> <p>An updated Primary Document when validating the Primary Document</p>
-------	--

Parameters Passed from Business Process to Service

The following table describes the parameters that are passed from a business process to the service:

Field	Description
certificateldentifier	Alias of a certificate public key. When used during signing, it indicates that the KeyInfo element must be included in the Signature. Only applicable when signing the Primary Document. Valid value is a valid alias name. Required.
ds:Transforms	Required Transforms to be used in the signing. If omitted, the enveloped-signature Transform will be used. Only applicable when signing the Primary Document. Valid value is a valid node.
incomingDoc	Node to be digitally signed. When not specified, it implies that the Primary Document needs to be signed. Valid value is a valid node.
nodeToSign	Required node to be signed. This node is found within the Primary Document. If not specified it implies that we wish to sign the whole document. Only applicable when signing the Primary Document. Valid value is a valid node name.
signCertificateldentifier	Alias of a private key of a certificate. Valid value is a valid alias name.

The incoming document must contain the prerequisites of the ebXML XML Digital Signature service. The following is an example node that is passed to the ebXML XML Digital Signature service for signature creation:

```
<ebXMLMessage>
<mime:message xmlns:mime="http://www.company.com/mime/v0.5">
<mime:header name="Content-Type">multipart/related<mime:parameter
name="start">ebxml-envelope@company.com</mime:parameter>
<mime:parameter name="type">text/xml</mime:parameter>
</mime:header>
<mime:header name="SOAPAction">ebXML</mime:header>
<mime:body>
<mime:message>
```

```

<mime:header name="Content-ID"><ebxml-envelope@company.com></mime:header>
<mime:header name="Content-Type">text/xml<mime:parameter
name="charset">UTF-8</mime:parameter>
</mime:header>
<mime:body>

<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">

<SOAP-ENV:Header
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:MessageHeader SOAP-ENV:mustUnderstand="1"
eb:id="ebxml-MessageHeader-company.com" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header
-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:From>
<eb:Role>http://www.company.com/roles/Sender</eb:Role>
</eb:From>
<eb:To>
<eb:Role>http://www.company.com/roles/Receiver</eb:Role>
</eb:To>
<eb:CPAId>CompanyID-CompanyID</eb:CPAId>
<eb:ConversationId>server::111z1:zzz999z9z:-1111</eb:ConversationId>
<eb:Service eb:type="string">FileTransfer-Sign</eb:Service>
<eb:Action>Receive</eb:Action>
<eb:MessageData>
<eb:MessageId>server::111z1:zzz999z9z:-1111</eb:MessageId>
<eb:Timestamp>2005-07-18T04:10:18Z</eb:Timestamp>
</eb:MessageData>
<eb:Description xml:lang="en-US">An ebXML Message.</eb:Description>
</eb:MessageHeader>

<eb:SyncReply SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
SOAP-ENV:mustUnderstand="1" eb:id="ebxml-SyncReply-company.com" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header
-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>

<eb:AckRequested SOAP-ENV:mustUnderstand="1" eb:id="ebxml-AckRequested-company.com"
eb:signed="true" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header
-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
</SOAP-ENV:Header>
<SOAP-ENV:Body/>
</SOAP-ENV:Envelope>
</mime:body>
</mime:message>
<mime:message>
<mime:header name="Content-ID"><ebxml-payload-0@company.com></mime:header>
<mime:header name="Content-Type">application/xml</mime:header>
<mime:body>
<doc:document-id
xmlns:doc="http://www.company.com/document-id">server:41114kd111rrrr4:-99zz</doc:doc
ument-id>
</mime:body>

```

```

</mime:message>
</mime:body>
</mime:message>
</ebXMLMessage>
</mime:message>
</mime:body>
</mime:message>
</ebXMLMessage>

```

The following parameters need to be passed to the service when it runs with the output message set to verifyRequest:

Field	Description
certificateIdentifier	Alias of a public key of a certificate. Valid value is a valid alias name.
documentID	Document-id of the MIME message to be certified. Valid value is a valid document id.
incomingDoc	Node that needs to have its digital signature verified. Valid value is a valid node.
removeSignature	Indicates if the validated document needs to have the Signature element removed. Only applicable when validating the Primary Document. Valid values are True and False.

The incoming document must contain the prerequisites of the ebXML XML Digital Signature service. The following is an example node that is passed to the ebXML XML Digital Signature service for signature validation:

```

<ebXMLMessage>
<mime:message xmlns:mime="http://www.company.com/mime/v0.5">
<mime:header name="Content-Type">multipart/related<mime:parameter
name="start">ebxml-envelope@company.com</mime:parameter>
<mime:parameter name="type">text/xml</mime:parameter>
</mime:header>
<mime:header name="SOAPAction">ebXML</mime:header>
<mime:body>
<mime:message>
<mime:header name="Content-ID"><ebxml-envelope@company.com></mime:header>
<mime:header name="Content-Type">text/xml<mime:parameter
name="charset">UTF-8</mime:parameter>
</mime:header>
<mime:body>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
<SOAP-ENV:Header
xmlns:eb="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:MessageHeader SOAP-ENV:mustUnderstand="1"
eb:id="ebxml-MessageHeader-company.com" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:From>

```

```

<eb:Role>http://www.company.com/roles/Sender</eb:Role>
</eb:From>
<eb:To>
<eb:Role>http://www.company.com/roles/Receiver</eb:Role>
</eb:To>
<eb:CPAId>company-company</eb:CPAId>
<eb:ConversationId>server::11c1:88888zzzz:-1111</eb:ConversationId>
<eb:Service eb:type="string">FileTransfer-Sign</eb:Service>
<eb:Action>Receive</eb:Action>
<eb:MessageData>
<eb:MessageId>server::11c1:88888zzzz:-1111</eb:MessageId>
<eb:Timestamp>2002-07-18T04:10:18Z</eb:Timestamp>
</eb:MessageData>
<eb:Description xml:lang="en-US">An ebXML Message.</eb:Description>
</eb:MessageHeader>

<eb:SyncReply SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
SOAP-ENV:mustUnderstand="1" eb:id="ebxml-SyncReply-company.com" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>

<eb:AckRequested SOAP-ENV:mustUnderstand="1" eb:id="ebxml-AckRequested-company.com"
eb:signed="true" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference Type="http://www.w3.org/2000/09/xmldsig#Object" URI="">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116"/>
<ds:XPath>
not(ancestor-or-self::node() [ @SOAP-ENV:actor="urn:oasis:names:tc:ebxml-msg:service:nextMSH" ] | ancestor-or-self::node() [ @SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next" ] ) </ds:XPath>
</ds:Transform>
<ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>+TTgggFZZZ+444t444DDffEEEddddd=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="cid:ebxml-payload-0@company.com"> <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>5SS4d44dGG1DD1DDddd3FFFee2GGGddd=</ds:DigestValue> </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
ZZzzZZzzZZzzZZzzzzZZzAAaaaEEaaaEEEEaaaaEEEaaaaDDDDddDDdddzzzzZZzzZZzz
zzAA33AAzzz44ZZaaZZZaaZZZaaZZZZ4eeEEE5WWwqqqEEEaaa4AAae5DDDDdEEEEddd2DDD
9Ik99R3Eeer444rrrFFF4694eee000333aaddii9991=
</ds:SignatureValue>
</ds:Signature>
</SOAP-ENV:Header>

```

```

<SOAP-ENV:Body/>
</SOAP-ENV:Envelope>
</mime:body>
</mime:message>
<mime:message>
<mime:header name="Content-ID"><ebxml-payload-0@company.com></mime:header>
<mime:header name="Content-Type">application/xml</mime:header>
<mime:body>
<doc:document-id
xmlns:doc="http://www.company.com/document-id">server:999z9:ggggh9g9g:-99zz</doc:doc
ument-id>
</mime:body>
</mime:message>
</mime:body>
</mime:message>
</ebxmlMessage>

```

Implementing the ebXML XML Digital Signature Service

To implement the ebXML XML Digital Signature service, complete the following tasks:

1. Activate your license for the ebXML XML Digital Signature service. See *Installing Application*.
2. Create an ebXML XML Digital Signature service configuration. See *Managing Services and Adapters*.
3. Configure the ebXML XML Digital Signature service. For information, see *Configuring the ebXML XML Digital Signature Service* on page 392.
4. Use the ebXML XML Digital Signature service in a business process.

Import a Key certificate for the ebXML XML Digital Signature service to reference in the business process. For the procedure, see *Importing a Key Certificate into a Business Process* on page 393.

Configuring the ebXML XML Digital Signature Service

To configure the ebXML XML Digital Signature service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
certificateIdentifier	The alias of a public key of a certificate. Valid value is a valid alias name.
incomingDoc	The node that needs to be digitally signed. Valid value is a valid node.
signCertificateIdentifier	The alias of a private key of a certificate. Valid value is a valid alias name.

Importing a Key Certificate into a Business Process

When importing a Key certificate, your BPML should use this alias to associate with the appropriate field (signCertificateIdentifier). The following BPML provides an example of how your BPML should look:

```
<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="incomingDoc" from="ebXMLMessage/mime:message"/>
<assign to="signCertificateIdentifier"
from="cpa/thisParty/signing-certificate-name/text()"/>
</output>
<input message="signResponse">
<assign to="ebXMLMessage"
from="signedEnvelope/signRequest/incomingDoc/node()"/>
</input>
</operation>
```

Business Process Example

The following example assumes that the nodes are found in the process data:

```
<process name="Test_Sign">
<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="incomingDoc" from="ebXMLMessage/mime:message"/>
<assign to="signCertificateIdentifier"
from="cpa/thisParty/signing-certificate-name/text()"/>
</output>
<input message="signResponse">
<assign to="ebXMLMessage"
from="signedEnvelope/signRequest/incomingDoc/node()"/>
</input>
</operation>
</process>
```

The following example shows how to sign the whole Primary Document and add the KeyInfo element with the Signature element by including the certificateIdentifier parameter in the signing request. The example also includes how to construct the ds:Transforms node:

```
<process name="TestSigning">
<sequence>
<assign to="temp/@Algorithm"
from="'http://www.w3.org/2000/09/xmlsig#enveloped-signature'"/>
<assign to="ds:Transforms/ds:Transform" from="temp/@*" />
<assign to="temp/@Algorithm" from="'http://www.w3.org/TR/1999/REC-xpath-19991116'"/>
<assign
to="temp/ds:XPath">count (/ETrade/DataArea/LCAdvice/descendant-or-self::node() |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/* |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/*/namespace:*) = count(. |
/ETrade/DataArea/LCAdvice/descendant-or-self::node() |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/* |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/*/namespace:*)
</assign>
<assign to="ds:Transforms/ds:Transform" from="temp/@* | temp/node()" append="true"/>
<!-- Do the Signing -->
```

```

<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="." from="*" />
<assign to="signCertificateIdentifier" from="'PrivKey'"/>
<assign to="certificateIdentifier" from="'PubKey'"/>
<assign to="ds:Transforms" from="ds:Transforms/node()" />
</output>
<input message="signResponse">
<assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

This BPML example shows how to sign a particular node in the Primary Document

```

<process name="TestSigning">
<sequence>
<assign to="temp/@Algorithm"
from="'http://www.w3.org/2000/09/xmldsig#enveloped-signature'"/>
<assign to="ds:Transforms/ds:Transform" from="temp/@*" />
<assign to="temp/@Algorithm" from="'http://www.w3.org/TR/1999/REC-xpath-19991116'"/>
<assign
to="temp/ds:XPath">count (/ETrade/DataArea/LCAdvice/descendant-or-self::node() |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/* |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/*/* |
/ETrade/DataArea/LCAdvice/descendant-or-self::node() |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/* |
/ETrade/DataArea/LCAdvice/descendant-or-self::*/*/*)
</assign>
<assign to="ds:Transforms/ds:Transform" from="temp/@* | temp/node()" append="true"/>
<!-- Do the Signing -->
<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="." from="*" />
<assign to="signCertificateIdentifier" from="'PrivKey'"/>
<assign to="nodeToSign" from="'TransactionResultResponse'"/>
<assign to="certificateIdentifier" from="'PubKey'"/>
<assign to="ds:Transforms" from="ds:Transforms/node()" />
</output>
<input message="signResponse">
<assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

This BPML example shows how to validate a signed Primary Document using a specified public certificate stored in GIS. The signature element will be removed from the Primary Document after the successful validation.

```

<process name="TestValidating">
<sequence>
<!-- Do the verification -->
<operation name="VerifyMessage">

```

```
<participant name="ebXMLDSig"/>
<output message="verifyRequest">
<assign to="." from="*"/>
<assign to="removeSignature" from="'true'"/>
<assign to="certificateIdentifier" from="'PubKey'"/>
</output>
<input message="verifyResponse">
<assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

EJB Adapter

The Enterprise Java Bean (EJB) adapter is comprised of the EJB Listener service and the GIS Adapter EJB (a J2EE application) that work together to enable you to launch a Application business process from a remote J2EE application server.

For this document a remote application server means one that is not associated with Application. The remote application server may reside on the same host as Application or not.

The following table provides an overview of the EJB Adapter Listener service:

System name	GISEJBAdapter
Graphical Process Modeler (GPM) category	This service is not used in business processes.
Description	<p>The EJB Adapter, using the EJB Adapter Listener service, defines a port on the Application host and listens for incoming requests from a remote application server. Based on the request, the service can:</p> <ul style="list-style-type: none">◆ Initiate (bootstrap) a business process◆ Pass a document and other data from the application server into process data of the initiated business process◆ Return a status about the business process◆ Return the results of the business process <p>The GIS Adapter EJB is deployed on a remote application server and is used to send requests to Application.</p>
Business usage	The EJB Adapter enables you to integrate programs on your existing application server with the business process transaction engine provided by Application.
Usage example	You create purchase orders on your legacy application server and you want to use Application to route the order for approval, translate the order into EDI, transfer the order to the vendor and update the legacy application with the status of the purchase order throughout that process. Use the GIS Adapter EJB to write a client program on your application server that sends the purchase order to Application, initiates a business process and requests status.
Preconfigured?	No
Requires third party files?	<p>Yes. One of the following J2EE application servers:</p> <ul style="list-style-type: none">◆ BEA® WebLogic® 7.0 SP2 & SP3◆ JBoss™ 3.2◆ IBM® WebSphere® 5.0.2
Platform availability	All supported Application platforms
Related services	None

Application requirements	<p>The following statements apply:</p> <ul style="list-style-type: none"> ◆ The Java Archive file <code>GISInvokerEJB_appServProductName.JAR</code> must be installed and configured on a WebLogic, WebSphere, or JBoss application server. See <i>Deploying GIS Adapter EJB on a WebSphere Application Server</i>, <i>Deploying GIS Adapter EJB on a WebLogic Application Server</i>, or <i>Deploying GIS Adapter EJB on a JBoss Application Server</i>. ◆ You must create the client program on the application server. The client program is responsible for sending the requests to Application.
Initiates business processes?	Yes. This service initiates a business process on demand from a remote application server.
Invocation	This service is inbound (bootstrapping) only, and will launch a business process on demand (non-pollled).
Business process context considerations	This service creates an initial business process context (WFC) with the documents and parameters provided by a client program at execution time.
Returned status values	<p>If the application server needs to obtain the status or the results of the Application business process, you can use the following two methods in the client program: <code>getStatus()</code> and <code>getResults()</code>.</p> <ul style="list-style-type: none"> ◆ Success – The EJB call to Application was successful. ◆ Error – The EJB call to Application was not successful.
Restrictions	<p>These restrictions apply:</p> <ul style="list-style-type: none"> ◆ The communication from the application server to Application is asynchronous only. ◆ The Adapter Listener service cannot be used in a business process. It is only used for initiating a business process.
Persistence level	Full Persistence
Testing considerations	<p>To test, do the following:</p> <ol style="list-style-type: none"> 1 Install and configure the EJB Adapter Listener service 2 Deploy the GIS Adapter EJB in the remote application server 3 Write a client program to send requests to Application. <p>See <i>EJB Client Program Example</i>.</p>

How the EJB Adapter Works

The following steps illustrate a synchronous communication between the remote application server and Application:

1. A legacy system running on a remote J2EE application server has a document that requires processing by Application.
2. A client program on the remote application server uses the GIS Adapter EJB to transfer the document and request that a specific business process be started. The program can also send requests for status, and, when the status indicates complete, a request for the resulting processed document.
3. The EJB Adapter Listener service receives the request, initiates the requested business process and passes the document into process data.

4. When the service receives the status request, it passes the business process status back to the remote application server. When the service receives a result request it passes the primary document back to the remote application server.

Implementing Communications with a Remote Application Server

To implement synchronous communication from a remote application server to Application two components must be set up:

The EJB Adapter Listener service that defines a port on the Application host.

The J2EE application, `GISInvokeEJB_appServProductName.jar`, that you install on the remote application server.

Complete the following process:

1. Create an EJB Adapter Listener service configuration.
2. Configure the EJB Adapter Listener service.
3. Create and enable a business process that you want to have initiated by the remote application server. For example, create a business process that translates documents from a legacy system on the remote application server.
4. Deploy the GIS Adapter EJB on the remote application server. See one of the following:
 - ◆ *Deploying GIS Adapter EJB on a WebSphere Application Server*
 - ◆ *Deploying GIS Adapter EJB on a WebLogic Application Server*
 - ◆ *Deploying GIS Adapter EJB on a JBoss Application Server*
5. Test the client program on the remote application server with the business process.

Configuring the EJB Adapter Listener Service

To configure the EJB Adapter Listener service, you must specify field settings in Application. There are no fields that must be defined in the Graphical Process Modeler (GPM).

Application Configuration

The following table describes the fields used to configure the EJB Adapter Listener service in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Listen Port (listenPort)	Port to which the remote application server sends requests. Be certain to select an available port for this parameter. Required.
Listen Hostname (ListenHostname)	Name or IP address of the host to which the remote application server sends requests. Optional.
Document Storage Type (docStorageType)	<p>Defines how the document will be stored in the system. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Note: For more information about document storage types, see <i>Managing Services and Adapters</i>.</p>

Deploying GIS Adapter EJB on a WebSphere Application Server

If your legacy tools reside in a WebSphere application server, follow these steps to install the J2EE application GIS Adapter EJB and deploy it in that server:

1. Copy the *install_directory/client/ejb/GISInvokerEJB.jar* file from the Application installation onto the machine that hosts your WebSphere application server.
2. Start the WebSphere server and go to the Administrative Console page.
3. Select **Install New Application** and enter the absolute path of the GISInvokerEJB.jar file.
4. Complete the installation procedure using the following guidelines:
 - ◆ Enter GISAdapterEJB for the Application Name
 - ◆ Enter GISAdapterEJB for the JNDI Name.
 - ◆ Update the access control list with the proper user information and be certain to disable the parameter Deny All.
5. Save the successful installation to the Master Configuration.
6. Navigate to **Environment > Naming > Name Space binding** and select **New**.
7. Configure the WebSphere environment using the following guidelines:
 - ◆ Obtain the name of the host that Application is installed on
 - ◆ Obtain the listenPort value used in the EJB Adapter Listener service configuration. This is the Listen Port parameter you used when configuring the service.

- ◆ Create a new Name Space binding using the Binding Type String for the Application host and port. Enter `host` for the Name in Name Space and your Application host name for the String Value. Enter `port` for the Name in Name Space and your service configuration `listenPort` value for the String Value.
8. Save the environment to the Master Configuration.
 9. You can now create a client program on your Websphere server that can initiate a Application business process.
See *EJB Client Program Example* for a sample Context Holder Object and a list of methods you can use in a client program and the properties they correspond to in Application.

Deploying GIS Adapter EJB on a WebLogic Application Server

Note: This section is not needed if you chose to integrate with WebLogic during the Application installation process.

If your legacy tools reside in a WebLogic application server, follow these steps to install the J2EE application GIS Adapter EJB and deploy it in that server:

1. Copy the *install_directory/client/ejb/GISInvokerEJB.jar* file from the Application installation onto the machine that hosts your WebLogic application server.
2. Start the WebLogic server and go to the Administrative Console page. The console page can be found at `http://host:weblogic_port/console`.
3. Select **Deployments** > **EJB** under `mydomain` from the left toolbar menu.
4. Select the directory into which you want to upload the EJB adapter.
5. Click **upload it through your browser**.
6. Select the absolute path of the `GISInvokerEJB.jar` file and upload the file. The file `GISInvokerEJB.jar` should now be listed in the directory list.
7. Click the **Select** link for `GISInvokerEJB.jar`.
8. Highlight your server and move it from Available Servers to Target Servers.
9. Select the servers onto which you want to deploy GIS Invoker EJB.
10. Complete the installation procedure using `GISAdapterEJB` for the Application Name. There should be a Completed message in the Status column and `GISAdapterEJB` should appear under EJB in the left toolbar menu.
11. You can now create a client program on your WebLogic server that can initiate a Application business process.

See *EJB Client Program Example* for a sample Context Holder Object and a list of methods you can use in a client program and the properties they correspond to in Application.

Note: You can bind the host and port to JNDI in the java program by the following method:

```
InitialContext env = (InitialContext) ctx.lookup("java:comp/env");
ctx.rebind("host", "hostname");
ctx.rebind("port", "listenerPort");
```


Deploying GIS Adapter EJB on a JBoss Application Server

Note: This section is not needed if you chose to integrate with JBoss during the Application installation process.

If your legacy tools reside in a JBoss application server, follow these steps to install the J2EE application GIS Adapter EJB and deploy it in that server:

1. Copy the *install_directory/client/ejb/GISInvokerEJB.jar* file from the Application installation into the deploy directory on the machine that hosts your JBoss application server.
2. Start the JBoss server and go to the Administrative Console page. The console page can be found at http://host:jboss_port/web-console.
3. Select **J2EE Domains > Manager > JBoss (<http://www.jboss.org/>) - 3.2.5 > GISInvokerEJB.jar > GISInvoker** from the JBoss Management Console menu. You can see the state of the GISInvoker Bean and confirm that the bean was successfully deployed to the JBoss application server.
4. You can now create a client program on your JBoss server that can initiate a Application business process.

Note: You can safely redeploy the application if it is already deployed. To undeploy it, just remove the archive from the *deploy* directory. You do not need to restart the server in either case.

See *EJB Client Program Example* for a sample Context Holder Object and a list of methods you can use in a client program and the properties they correspond to in Application.

Note: You can bind the host and port to JNDI in the java program by the following method:

```
InitialContext env = (InitialContext) ctx.lookup("java:comp/env");
ctx.rebind("host", "hostname");
ctx.rebind("port", "listenerPort");
```

EJB Client Program Example

Before you can create a client program that will initiate a Application business process you must:

Configure the EJB Adapter Listener service in Application.

Create and check in the business process you want to initiate.

Install and configure the GIS Adapter EJB on your remote J2EE application server.

The following lists Application-specific methods you can use in the client program:

Purpose	Method
//Execute business process	<code>ContextHolder rch =.ejb.submitContext(ch, "login", "password");</code>
//Obtain business process status	<code>int wf_status =.ejb.getState(rch, "admin", "password");</code>
//Obtain process data	<code>ContextHolder con= .ejb.retrieveContext(rch, "login password</code>

The following is an example of a program you could create on your application server that would work with Application.

Encoding Conversion Service

The following table provides an overview of the Encoding Conversion service:

System name	EncodingConversionType
Graphical Process Modeler (GPM) categories	All Services, Translation
Description	<p>Converts a file from one character encoding to another as part of an Application business process. A <i>character encoding</i> is a representation of data in a particular character set. A character set is a list of characters (letters, numbers, and symbols) that are recognized by computer hardware and software. Examples of encoding sets are American Standard Code for Information Interchange (ASCII), eight-bit Unicode Transformation Format (UTF-8), and Simplified Chinese (GBK).</p> <p>Caution: The encoding you are converting from must have characters similar to the encoding you are converting to.</p>
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	The encoding you are converting from must have characters similar to the encoding you are converting to.
Invocation	Runs as part of a business process.

Implementing the Encoding Conversion Service

To implement the Encoding Conversion service, complete the following tasks:

1. Activate your license for the Encoding Conversion service.
2. Create an Encoding Conversion service configuration.
3. Configure the Encoding Conversion service.
4. Use the Encoding Conversion service in a business process.

When creating a business process, determine whether to extract the converted file from Application to a specific file location. To extract the file, follow the Encoding Conversion service with a File System adapter configuration in your business process.

Configuring the Encoding Conversion Service

To configure the Encoding Conversion service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
input_encoding	Encoding of the input file. Required.
output_encoding	Encoding to which you want to convert the file. Required.

End Transaction Service

The following table provides an overview of the End Transaction service.

System name	End Transaction Service
Graphical Process Modeler (GPM) categories	All Services
Description	Ends a multi-step transaction in a business process.
Business usage	<p>Use this service to mark the end of a multi-step transaction in a business process. All of the steps that occur between the Begin Transaction service and End Transaction service are part of one transaction and are committed or rolled back at the same time.</p> <p>This service should be used with the Begin Transaction service.</p>
Usage example	<p>Consider a customer ordering items online and using a shopping cart. The merchant's business process:</p> <ol style="list-style-type: none">1. Gets a document containing shopping cart information (customer information, product details).2. Inserts the customer information to the customer table.3. Inserts product details to the invoice table (product_id, quantity purchased).4. Updates the quantity of the product from the inventory table as the products were sold. <p>If step 4 fails (update the quantity), but steps 1 and 2 were committed. The merchant's inventory table is now inaccurate, because the items were reported as sold, but the items were not deducted from the inventory table. The Begin Transaction service and End Transaction service make it possible to have a transaction where all the steps are committed successfully or all the steps are rolled back.</p>
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	The End Transaction service should be used with the Begin Transaction service.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	No

Returned status values	Possible status values: <ul style="list-style-type: none">◆ 0 – Success◆ 1 – Error
Restrictions	N/A
Testing considerations	N/A

How the End Transaction Service Works

Use the End Transaction service to end a multi-step transaction.

Business Process Example

In the following business process example, an online shopping cart is featured. The business process:

1. Gets a document containing shopping cart information (customer information, product details).
2. Inserts the customer information to the customer table.
3. Inserts product details to the invoice table (product_id, quantity purchased).
4. Updates the quantity of the product from the inventory table as the products are sold.

For example, if step 4 fails (update the quantity), but steps 1 and 2 were committed. The merchant's inventory table is now inaccurate, because the items were reported as sold, but the items were not deducted from the inventory table. The Begin Transaction service and End transaction service make it possible to have a transaction where all the steps are committed successfully or all the steps are rolled back.

Implementing the End Transaction Service

To implement the End Transaction service for use in a business process:

1. Use the End Transaction service in a business process.
2. Use the Begin Transaction service to begin the transaction.

Output from Business Process to Service

The following table contains the parameter passed from the business process to the End Transaction service:

Parameter	Description
End_Transaction	Ends the transaction. Valid value is true (default) or false. Required.
Rollback_Transaction	Sets the transaction to rollback. Valid value is true or false (default). Optional. Business process logic may be used to determine that a transaction should not be committed. By calling an instance of this service with this parameter set to "TRUE", the transaction can be rolled back.

Execution Control Service

The Execution Control service allows for the business process to dynamically change various execution parameters, primarily for performance. The following table provides an overview of the Execution Control service:

System name	ExecutionControlService
Graphical Process Modeler (GPM) categories	All Services, Process Controls
Description	The Execution Control service allows for the business process to dynamically change various execution parameters, primarily for performance.
Business usage	<p>Use this service to dynamically affect the execution of a business process. A business process can be:</p> <ul style="list-style-type: none">◆ Assigned to a different queue◆ Assigned a different priority within its queue◆ Assigned an absolute deadline (in milliseconds, as provided by the Timestamp Utility service) <p>For example, a received document will be held for batch processing, but a functional acknowledgement for it should be expedited. You can set the spawned business process for the functional acknowledgement at a higher priority than the batch process.</p>
Usage example	<p>At run time, this service can dynamically move within a business process to a lower or higher queue than the one assigned at check in time. Moving to a lower queue can free up threads for the highest priority processes. Additionally, the Execution Control service allows configuring for fewer steps in the process to be completed before re-entering the queue for an available thread in the workflow engine.</p> <p>For example, you can give a business process with a service level agreement time window processing priority by configuring it to hold onto its thread for many steps. In addition, you can move it up in the priority queue at a certain step. Conversely, you can configure a lower priority process to do very few steps before returning to the workflow engine for an available queue, thereby freeing resources for higher priority processing.</p>
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	The business process using this service should have appropriate queue priorities assigned.
Initiates business processes?	This service does not initiate business processes. It works within a business process to adjust the execution of the steps in its business process, and/or change its priority queue.
Invocation	Does not invoke a business process

Business process context considerations	This service works within the workflow context to dynamically reassign: <ul style="list-style-type: none"> ◆ Process Execution Priority ◆ Execution Queue Name ◆ Execution Cycle Size ◆ Deadline
Returned status values	<ul style="list-style-type: none"> ◆ Success – Execution was successful ◆ Error – Unexpected parameter failure (for unparsable data)
Restrictions	None
Persistence level	None

Implementing the Execution Control Service

To implement the Execution Control service, complete the following tasks:

1. Create an Execution Control service configuration.
2. Configure the Execution Control service.
3. Use the Execution Control service in a business process.

Configuring the Execution Control Service

To configure the Execution Control service, you must specify settings for the following fields in the Graphical Process Modeler (GPM):

Field	Description
deadline	Deadline time by which the business process should be executed, in milliseconds as provided by the Timestamp Utility service. Optional.
executionCycleSize	Number of steps AE will cycle. Required. Value values are 1 - 2147483647. Default is 10.
mandatoryNode	Cluster only. The specific node on which the business process must run. Optional. Specify any valid node number. You will receive an error if the specified node is not available. Note: The Application initializes a business process workflow on any available node before moving the business process to the node specified in mandatoryNode.
preferredNode	Cluster only. The node on which you would like the business process to run. Optional. Specify any valid node number. If the specified node is not available, the service will use another available node.
priority	Priority suggestion within the queue to place a step within a business process. Required. Valid values are 1 - 9. Default is 4.
queueName	The queue to be placed into. Optional. Valid values are 1 - 9.

Field	Description
yield	<p>Allow the business process to yield its thread to a higher priority process. After the yield, it is placed back in the queue. Required. Valid values:</p> <ul style="list-style-type: none">◆ true◆ false (default) <p>Note: In order for changes to other settings, such as executionCycleSize, to take effect immediately, yield must be set to true.</p>

The fields configured in the GPM are passed from the business process to the Execution Control service.

Business Process Example

The following sample BPML demonstrates the syntax and usage of the Execution Control Service:

Export Service

The following table provides an overview of the Export service:

System name	ExportService
Graphical Process Modeler (GPM) category	All Services
Description	<p>The Export service automates exporting resources from the application.</p> <p>You specify the resources to be exported and the various options in the Export service input XML file. Alternatively, you can specify a resource tag (which defines all the resources to be exported) in the input XML file. Resource tags can be created using the Admin Console. See <i>Using Proper Syntax</i> for syntax information.</p> <p>You can export resources to an XML file or to an installable bundle. If you want to export resources to be used on a new installation of your application, use an installable bundle. You can then load the installable bundle as part of the new system installation. See <i>Using Proper Syntax</i> for syntax information.</p> <p>The input XML file can also be used to specify the export type: standard or advanced.</p> <ul style="list-style-type: none">◆ Standard export copies non-versioned resources and the default version of versioned resources.◆ Advanced export copies non-versioned resources and enables you to choose to export just default versions or all versions of versioned resources. <p>See <i>Using Proper Syntax</i> for syntax information.</p> <p>You can specify regular expressions to filter and selectively export the desired resources. See <i>Regular Expressions</i> for information.</p> <p>When exporting trading partner identities, PGP profiles, service configurations, and communities, you must use a security context utility to store a passphrase in the database. The passphrase is then used to encrypt the identity information when it is exported to the output XML file. See <i>Using the Security Context Utility</i> for information.</p>
Business usage	The Export service can be used with the scheduler to automate the process of exporting application resources. You define the resources to be exported in an XML file.
Usage example	<p>The Export service is typically used in the following situations:</p> <ul style="list-style-type: none">◆ In a virtual VAN environment, trading partners will synchronize their business resources in an automated fashion by using a combination of export and import services.◆ Certain application resources may need to be exported and sent to SCI customer support to assist in troubleshooting problems.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms for your application
Related services	Import service
Application requirements	None

Initiates business processes?	No
Invocation	Generally, this service is invoked by the Scheduler.
Business process context considerations	None
Returned status values	<p>Success—Successfully exported</p> <p>Error—there are two basic errors:</p> <ul style="list-style-type: none"> ◆ There is no Primary Document, this service operates on the primary document ◆ Set passphrase using SecurityContext program for security context
Restrictions	None
Persistence level	Full
Testing considerations	Use the Export option from the Deployment > Resource Manager menu to export a set of resources. Save the status report from that export. Next, use the Export service to export the same set of resources. Compare the status reports for the two exports to determine if the Export service is successfully exporting all of the resources you specified.

How the Export Service Works

The Export service works based on an XML input file that you create. The input file must contain the context and identity values from a security context (only if trading partner identities are included), whether or not to export information based on resource tags, the resource tags or names, the output type (XML or installable bundle), and the export type (standard or advanced). It can contain much more information, however. There are many parameters and expressions that can be used to select just the resources you want. See *Regular Expressions* and *Options* for more information.

Example Business Scenario

Your company has a trading partner community called Suppliers. You want to update your company's trading partner profile and send the suppliers a new service configuration to use when sending you data. To accomplish this:

1. Create a resource tag (Supplier_Update) that will contain the trading partner profile and the service configuration.

From the Deployment menu, select **Resource Manager > Resource Tags > New Resource Tag**. Select the resources to associate.

2. Create a security context for this export operation:

```
install_dir\bin>securityContext.sh set Suppliers dec01 leopard
```

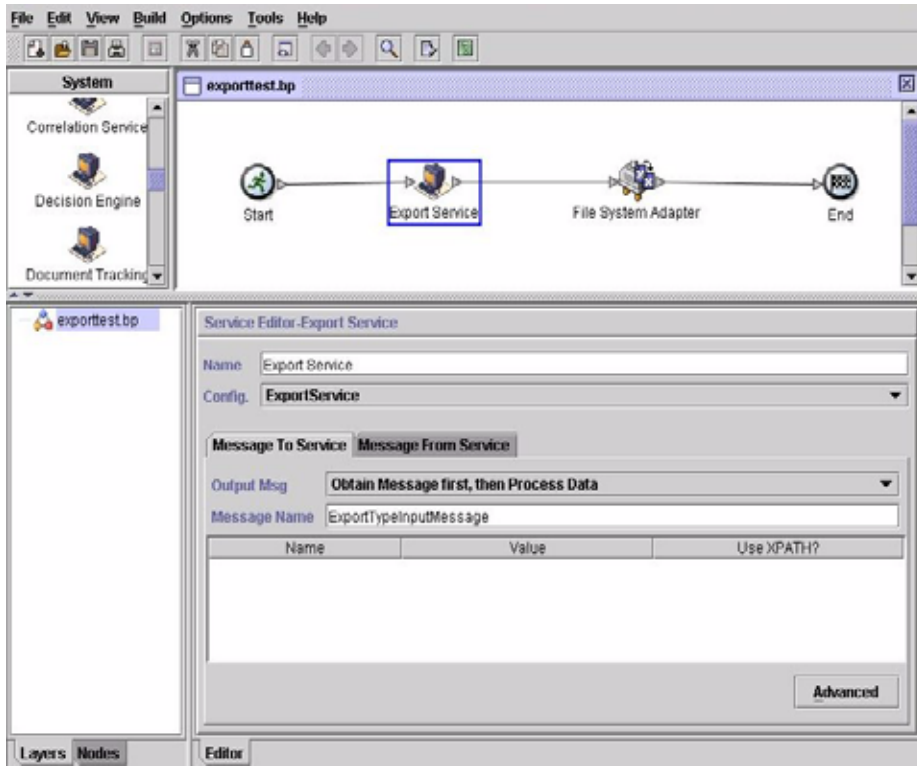
See *Using the Security Context Utility*.

3. Create a well-formed XML file to be used as input for the Export service. To test that the XML file is well-formed, open the file with Internet Explorer. Internet Explorer will open an XML file only if it is well-formed:

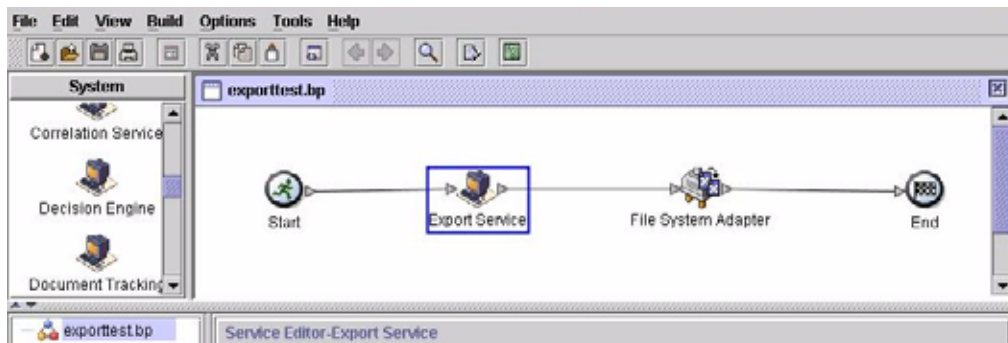
```
<?xml version="1.0"?>
<ExportConfiguration>
<OutputFormatType>XML Document</OutputFormatType>
<ExportResourcesBasedOnTagName>Yes</ExportResourcesBasedOnTagName>
<Tags>
<Tag Name="Supplier_Update" />
</Tags>
<ExportType>Standard</ExportType>
<Passphrase>
<SecurityContext>Suppliers</SecurityContext>
<SecurityIdentity>dec01</SecurityIdentity>
<ExportCertificate>Yes</ExportCertificate>
</Passphrase>
</ExportConfiguration>
```

4. Create an Export service configuration and a File System adapter configuration to use in your business process. In the File System adapter configuration, specify the extraction folder (where the File system adapter will place the export file at the end of the business process).

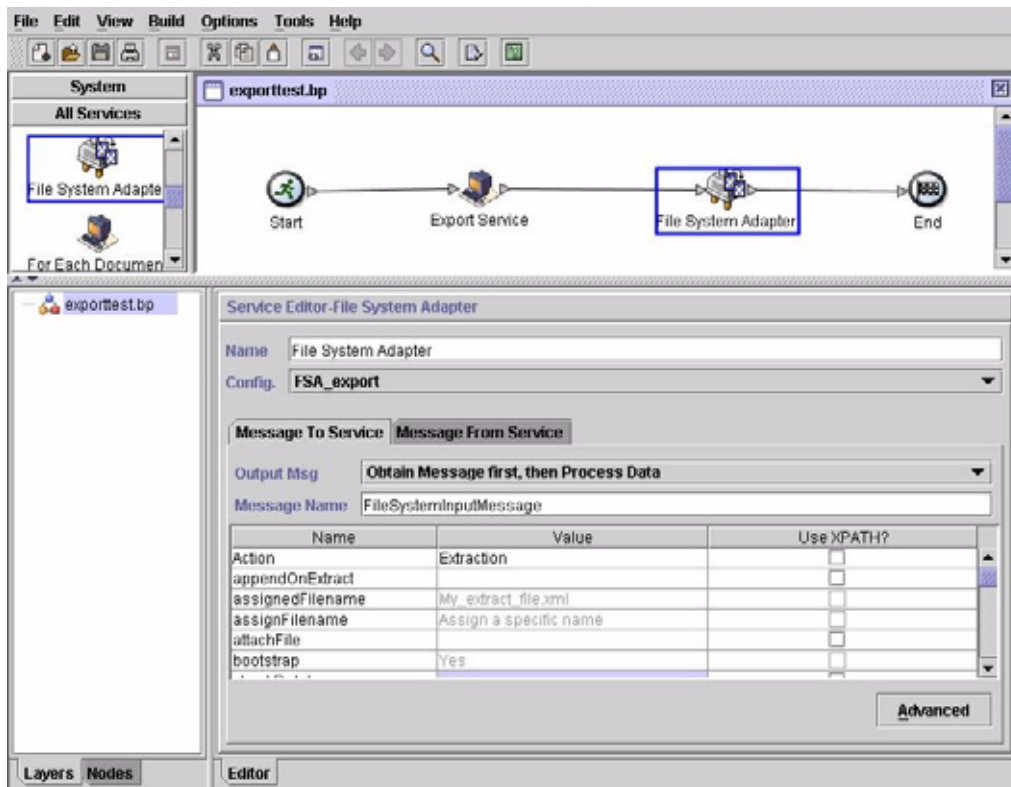
5. Create a business process to export the files. The Export service and File System adapter will be used to export the resources and then extract the export file to a location on your system:



To enter the information for the Security context, click **Advanced** and add two entries, Context and Identity. Enter the values for each from the security context you created earlier:



Use the File System adapter configuration you created earlier in this business process. Select Extraction as the action in the GPM:



Implementing the Export Service

To implement the Export service, complete the following tasks:

1. Identify the resources to be exported. If there are several types of resources (maps, services, trading partner information, for example), you can create a resource tag that includes all of the resources.
For a list of resources that can be exported, see *Resources That Can Be Exported*.
2. Create a security context and passphrase for this export operation, if required for the type of data you are exporting. For information, see *Using the Security Context Utility*.
3. Create an XML file to use as input for the Export service. This file defines the resources and type of export. For information, see *Creating an Input XML File*.
4. Create an Export service configuration. See *Managing Services and Adapters*.
5. If you want to have the export file placed in a folder on your system, create a File System adapter configuration. Specify the folder to which the file should be extracted. For XML output, specify .xml as the file extension.
6. Use the Export service and File System adapter in a business process.

Example Input XML Files

The following examples show how an XML file can be used as input for the Export service. The first example illustrates using resource tags; the second example illustrates using resource names.

Example 1—Using Resource Tags

An easy way to create an export file is to create resource tags in the Admin Console and write an input file that refers to them. The file can include one or more resource tags. This example exports all the resources identified by two tags:

```
<?xml version="1.0"?>
<ExportConfiguration>
<OutputFormatType>XML Document</OutputFormatType>
<ExportResourcesBasedOnTagName>Yes</ExportResourcesBasedOnTagName>
<Tags>
  <Tag Name="bp1"/>
  <Tag Name="maps1"/>
</Tags>
<ExportType>Standard</ExportType>
<Passphrase>
<SecurityContext>secret1</SecurityContext>
<SecurityIdentity>secret2</SecurityIdentity>
<ExportCertificate>Yes</ExportCertificate>
</Passphrase>
</ExportConfiguration>
```

Example 2—Using Resource Names

The following example specifies that resource names are used, rather than resource tags, and uses inclusions and exclusions to customize the data selected for the export:

```
<?xml version="1.0" ?>
<ExportConfiguration>
  <OutputFormatType>XML Document</OutputFormatType>
  <ExportResourcesBasedOnTagName>No</ExportResourcesBasedOnTagName>
  <ExportType>Standard</ExportType>
  <Resources>
    <Resource>
      <Name>Business Process</Name>
      <PatternSet>
        <include>*</include>
        <exclude>AdvanceShipment</exclude>
      </PatternSet>
    </Resource>
    <Resource>
      <Name>Trading Partner Data</Name>
      <SubResources>
        <SubResource>
          <Name>Contracts</Name>
          <PatternSet>
            <include>pipContract3A12_initiator</include>
          </PatternSet>
        </SubResource>
        <SubResource>
```

```
<Name>Identities</Name>
<PatternSet>
  <include>111111111</include>
  <include>999999999</include>
</PatternSet>
</SubResource>
</SubResources>
</Resource>
</Resources>
<Passphrase>
  <SecurityContext>CompanyA</SecurityContext>
  <SecurityIdentity>Apr15</SecurityIdentity>
  <ExportCertificate>Yes</ExportCertificate>
</Passphrase>
</ExportConfiguration>
```

Using the Security Context Utility

The following types of data require a security passphrase when exported:

- Trading Partner Identities

- Trading Partner Envelopes

- PGP profiles

- Service configurations

- Communities

- Certain SSH resources: Host Identity Keys, User Identity Keys, and SSH Remote Profiles

- Web services

- Security tokens

- Proxy netmaps

- System digital certificates

The Export service works with the Security Context utility in the application. The utility is called `securityContext.sh` (for Unix) or `securityContext.cmd` (for Windows). It is located in the bin directory of your application installation. The security context utility can be used to store the passphrase and to retrieve it. The security context and security identity values are used to retrieve the associated passphrase. The passphrase is used to encrypt sensitive information (such as trading partner identities) in the export file.

The following information is used in the examples in this section:

- Context = CompanyA

- Identity = Apr15

- Passphrase = secret_phrase

When the utility is used to store the passphrase, provide the following three parameters:

```
securityContext.sh set CompanyA Apr15 secret_phrase
```

To check on the existence of a passphrase, use the following command:

```
securityContext.sh get CompanyA Apr15
```

The retrieve the list of contexts, use the following command:

```
securityContext.sh list_context
```

The context and identity need to be specified in the input XML file to direct the Export service to retrieve the appropriate passphrase from the database, as shown in the following example:

```
<Passphrase>
  <SecurityContext>CompanyA</SecurityContext>
  <SecurityIdentity>Apr15</SecurityIdentity>
  <ExportCertificate>Yes</ExportCertificate>
</Passphrase>
```

Creating an Input XML File

In the XML files you create for input to the Export service, you can customize export operations by using the expressions and various options described in this section.

Using Proper Syntax

This section illustrates the correct syntax to use when specifying resource tags, output types and standard or advanced exports.

Use the following syntax when referencing a resource tag:

```
<Tags><Tag Name="Test2" /></Tags>
```

Use the following syntax when specifying the output type for the export:

- ◆ For Install Bundle: `<OutputFormatType>Install Bundle</OutputFormatType>`
- ◆ For XML Document: `<OutputFormatType>XML Document</OutputFormatType>`

Use the following syntax when specifying the export type:

- ◆ For Standard: `<ExportType>Standard</ExportType>`

The following example shows a sample standard export file:

```
<ExportType>Standard</ExportType>
<Resources>
  <Resource>
    <Name>Maps</Name>
    <PatternSet include= "^acme"/>
  </Resource>
</Resources>
```

- ◆ For Advanced: `<ExportType>Advanced</ExportType>`

The following example shows an advanced export that specifies all versions of versioned resources for map names that include acme should be exported:

```
<ExportType>Advanced</ExportType>
<Resources>
  <Resource>
    <Name>Maps</Name>
    <PatternSet include= "^acme"/>
    <AllVersions>true</AllVersions>
  </Resource>
</Resources>
```

Regular Expressions

You can use regular expressions to filter and selectively export resources such as business processes, contracts, and identities.

The following example XML fragment could be used in an XML file to export all business processes whose names begin with wssd:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet include= "^wssd"></PatternSet>
</Resource>
```

If using multiple includes, they are handled on separate lines, as shown in this example:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet>
    <include>^[a-m]</include>
    <include>wssd</include>
  </PatternSet>
</Resource>
```

The following example exports all business processes whose names begin with wssd except those whose names end with enu:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet include= "^wssd" exclude= "enu$"></PatternSet>
</Resource>
```

Options

You can customize exports using the options described in this section:

Characters

The following table contains the characters that you can use with the Export service:

Character	Description
UnicodeChar	Matches any identical unicode character
\	Used to quote a meta-character (like '*')
\\	Matches a single '\' character

Character Classes

The following table contains the character classes that you can use with the Export service:

Class	Description
[abc]	Simple character class
[a-zA-Z]	Character class with ranges
[^abc]	Negated character class

The following example exports all business processes whose names begin with letters A through M:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet include= "[a-m]"></PatternSet>
</Resource>
```

The following example exports all business processes whose names begin with letters A through C:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet include= "[abc]"></PatternSet>
</Resource>
```

Portable Operating System Interface Character Classes

The following table contains the standard Portable Operating System Interface (POSIX) character classes that you can use with the Export service:

Class	Description
[:alnum:]	Alphanumeric characters.
[:alpha:]	Alphabetic characters.
[:blank:]	Space and tab characters.
[:cntrl:]	Control characters.
[:digit:]	Numeric characters.

Class	Description
[[:graph:]]	Characters that are printable and are also visible. (A space is printable, but not visible, while an `a` is both.)
[[:lower:]]	Lower-case alphabetic characters.
[[:print:]]	Printable characters (characters that are not control characters.)
[[:punct:]]	Punctuation characters (characters that are not letter, digits, control characters, or space characters).
[[:space:]]	Space characters (such as space, tab, and formfeed, to name a few).
[[:upper:]]	Upper-case alphabetic characters.

The following example exports all business processes whose names contain a numeric character:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet include= "[[:digit:]]"></PatternSet>
</Resource>
```

The following example exports all business processes whose names begin with a lower-case character:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet include= "^[:lower:]"></PatternSet>
</Resource>
```

Logical Operators

The following table contains the logical operators that you can use with the Export service:

Operator	Description
AB	Matches A followed by B
A B	Matches either A or B
(A)	Used for subexpression grouping
(?:A)	Used for subexpression clustering (similar to grouping but no backrefs)

The following example exports all business processes whose names begin with letters E, M, or W:

```
<Resource>
  <Name>Business Process</Name>
  <PatternSet>
    <include>^e|^m|^w</include>
  </PatternSet>
</Resource>
```

If you're searching for non-ASCII characters, such as é or lå, save the input file as UTF-8.

Resources That Can Be Exported

The following list contains the types of resources that can be exported using the Export service:

Application configurations

Business processes

Communities

Contracts

Control numbers

Code lists

Documents

ebXML BPSS

ebXML CPA

Group permissions

Groups

HTML files

Identities

Images

Javascript files

JSP files

Mailbox META

Mailbox routing

Mailboxes

Maps

Other files

Perimeter servers

Permissions

Property files

Schedules

Schemas

SCI packages

Service configurations

Stylesheets

XML files

Xslts

User groups

User permissions

Users

Web services

WSDL

File System Adapter (Build 4300 - Build 4324)

The following table provides a high-level overview of the File System adapter:

System name	FileSystem
Graphical Process Modeler (GPM) category	All Services
Description	Collects (imports) files from the file system and extracts (exports) files to the file system. The collected file becomes the primary document in a business process for file collection. A primary document is input to the File System adapter for file extraction.
Business usage	To read files from disk or write files to disk.
Usage example	Commonly used at the beginning of a business process to bootstrap a workflow by reading one or more files from disk and starting a business process. Another common use is to write files to disk for some external purpose. Note: The term <i>bootstrap</i> is used in the Graphical Process Modeler to indicate that the File System adapter is used to start a business process after file collection.
Preconfigured?	No
Requires third party files?	None
Platform availability	All supported platforms for Application
Related services	None
Application requirements	None
Initiates business processes?	Yes, if you define a business process to start when you configure the File System adapter. The business process starts once files are collected.
Invocation	Normally, only the extraction side of the File System adapter is used in a business process, because you can configure the collection side of the File System adapter to start (“bootstrap”) a business process. However, you can include the File System adapter directly in a business process to perform an explicit collection of files.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – File System collection or extraction was successful.◆ Error – File System collection or extraction was unsuccessful.
Restrictions	You must have read access to files and directories for file collection and write access to directories for file extraction.
Persistence level	System Default (Full)

Testing considerations The best (and easiest) way to test this adapter is to set up a business process that only performs a file system extraction and specify that business process as the initial workflow to start (bootstrap).

How the File System Adapter Works

Use the File System adapter to collect (import) files from a file system into a business process and extract (export) files from a business process to a file system. You can configure the File System adapter to start a business process after files are collected from the file system or include the File System adapter in a business process flow. In addition, you can schedule the File System adapter to run at specific time intervals.

You can create multiple File System adapter configurations, one for each of several collection folders. Alternatively, you can use a single File System adapter configuration to point to different directories by specifying the directories for file collection and extraction explicitly in a business process. See *Usage Examples*.

The following sections describe a business scenario in which you could use the File System adapter, along with some sample solutions.

Business Scenario

Your company receives a purchase order from a trading partner in EDI file format and the file is stored on the internal file system. You need to translate the EDI file into XML format and write the translated file to a local directory.

Business Solution Example

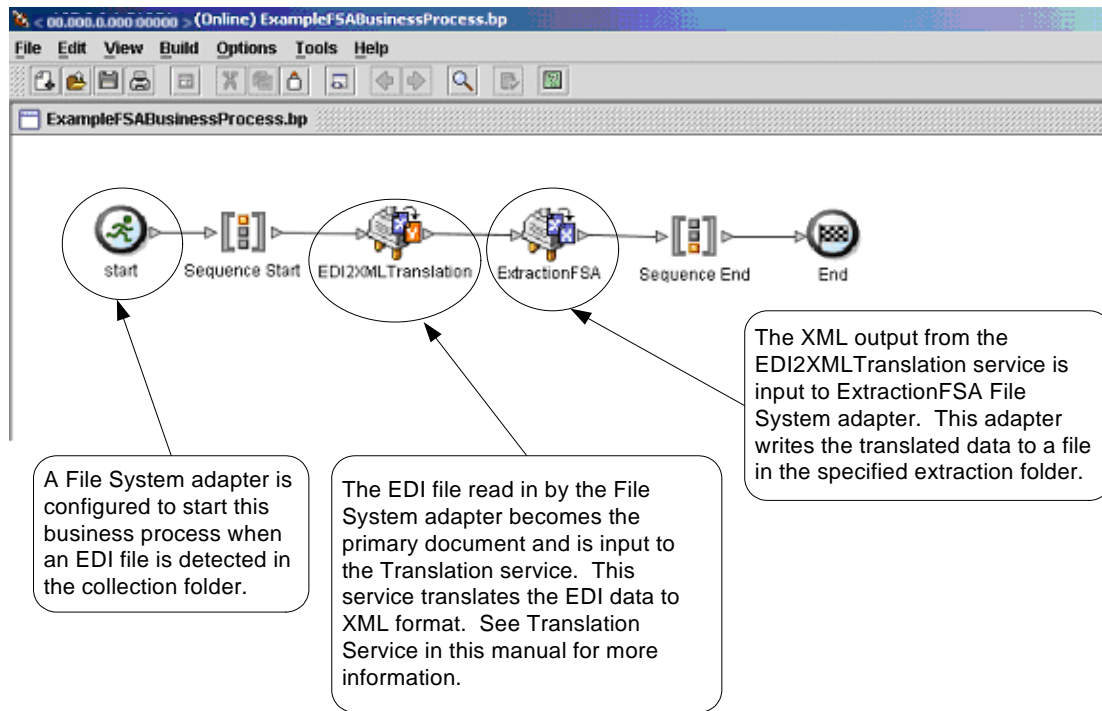
The following approaches are used to solve the above business scenario.

1. Configure a File System adapter instance to be included in a business process to perform a file extraction.
2. Create a business process that translates the EDI file to XML format and then uses the File System adapter instance configured above to extract the resultant XML data to the file system
3. Configure a separate File System adapter instance to start the business process created in the previous step after an EDI file is detected in the collection folder. This File System adapter instance is also scheduled to run at 30-minute intervals.

This business solution is described for both the Graphical Process Modeler (GPM) and the Business Process Modeling Language (BPML).

Graphical Process Modeler (GPM) Example

The following example shows a simple solution to the above business scenario using the GPM.



Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML.

```
<process name="ExampleFSABusinessProcess">
  <sequence name="Sequence Start">
    <operation name="EDI2XMLTranslation">
      <participant name="Translation"/>
      <output message="TranslationTypeInputMessage">
        <assign to="exhaust_input">YES</assign>
        <assign to="map_name">ExampleP0850</assign>
        <assign to="output_to_process_data">NO</assign>
        <assign to="validate_input">NO</assign>
        <assign to="validate_input_against_dtd">NO</assign>
        <assign to="validate_output">NO</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="ExtractionFSA">
      <participant name="ExampleExtractionFSA"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```



Implementing the File System Adapter

You can implement the File System adapter in three ways:

- Collect files within a business process.
- Extract files to the file system from a business process.
- Collect files and then start a new business process.

The information in this section applies to all three implementations.

Before you begin to implement a File System adapter, you need to collect the following information:

- The name of the business process (if the adapter is to start a business process)
- The directory path from which files are collected
- The directory path to which files are extracted

Process Overview

To implement the File System adapter, complete the following tasks:

1. Create a File System adapter configuration.
2. Configure the File System adapter.

3. Create a business process to run after the File System adapter collects files, or create and enable a business process that includes the File System adapter (collecting or extracting files).
4. Test the business process and the adapter.
5. Run the business process.

Configuring the File System Adapter

To create a File System adapter configuration, specify field settings in Application and in the GPM.

File System Configuration


The following table describes the fields used to configure the File System adapter in the Application.

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a group	Group to associate with the adapter. Valid values: <ul style="list-style-type: none"> ◆ None – No group is selected ◆ Create New Group – Allows the creation of a new group ◆ Select Group – Select from a list of available groups
Collection folder (collectionFolder)	The name of the folder or subfolder on the same computer where Application is installed and where it collects (or picks up) files as part of a business process. If the path for the folder is not included as part of the name, the folder is assumed to be in the Application working directory. Required. Note: <ul style="list-style-type: none"> ◆ The deleteAfterCollect parameter in the GPM defaults to Yes. If you do not change the default value to No, files that are collected are deleted from the Collection Folder. The File System adapter does not copy the files it collects for processing. See <i>Graphical Process Modeler Configuration</i> for information about the deleteAfterCollect parameter. ◆ The collectionFolder parameter is read-only in the GPM. However, you can override this parameter using BPML.

Field	Description
Filename filter (filter)	<p>Collect only files that match a specified filter within the collection folder. Optional.</p> <p>Examples include:</p> <ul style="list-style-type: none"> ◆ *.txt (collects only .txt files) ◆ *.dat (collects only .dat files) ◆ EDI.* (collects only files named EDI with any file extension) ◆ EDI.txt (collect only files named EDI with a file extension of .txt) <p>Note: If there are multiple files in the collection folder and you leave this field blank, one of the following occurs:</p> <ul style="list-style-type: none"> ◆ If the adapter is configured to start a business process, it processes all files placed in the collection folder. ◆ If the adapter is within a business process, it collects only the first file in the collection folder. <p>Note: If you specify this option using the File System adapter configuration, you cannot override the value using the GPM filter parameter. However, you can override this parameter using BPML.</p>
Collect files from subfolders within and including the collection folder? (useSubFolders)	<p>Whether to scan for files in subfolders of the collection folder. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Collects files in the specified folder and all subfolders. ◆ No – Collects files in the specified folder only. <p>Note: This parameter is read-only in the GPM.</p>
Use the absolute file path name for the document name? (keepPath)	<p>Whether to keep the absolute path name of the files collected when assigning the document name. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – The absolute file path name is kept with the document in the business process. Choose this value if your business process requires the path information to precede the file name. ◆ No – Only the file name is kept with the document in the business process. <p>Note: An absolute path is a path that points to the same location regardless of the working directory or combined paths. It is usually written in reference to a root directory. For example, c:\dir1\subdir1\somefile.txt (Windows) and /home/dir1/subdir1/somefile.txt (UNIX) are examples of absolute paths to the file somefile.txt.</p> <p>Note: This parameter is read-only in the GPM.</p>

Field	Description
Start a business process once files are collected? (bootstrap)	<p>Whether to start a business process using the File System adapter after files are collected. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Starts the business process specified from the business process drop-down list. <p>Note: An instance of the business process is started for every file that matches the filtering criteria specified for file collection until the number of threads specified on the <code>maxThreads</code> parameter is reached. See <i>Graphical Process Modeler Configuration</i> for information about the <code>maxThreads</code> parameter.</p> <ul style="list-style-type: none"> ◆ No – No business process will be started. <p>Note: This parameter is read-only in the GPM.</p>
Business Process (initialWorkflowId)	<p>The business process to start after files are collected. Required when <i>Start a business process</i> is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Name of the business process to start ◆ Not Applicable <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify a business process using the configuration, you cannot override this value using the GPM <code>initialWorkflowId</code> option. If you select Not Applicable, a business process can be selected in the GPM. In either case, you can override this parameter using BPML.</p>
Document storage type (docStorageType)	<p>Defines how the document will be stored in the system. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM <code>DocStorageType</code> option. However, you can override this parameter using BPML.</p> <p>Note: For more information about document storage types, see <i>Selecting a Document Storage Method for Bootstrap Adapters</i> on page 20.</p>
Obscure File Contents? (obscure)	<p>Specifies whether to obscure the file contents when collecting. Does not work with “attachFile” or “importFile”.</p> <ul style="list-style-type: none"> ◆ Yes – File contents will be obscured ◆ No – File contents will not be obscured <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM <code>Obscure</code> option. However, you can override this parameter using BPML.</p>

Field	Description
User Parameter 1 (userParm1)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm1. For more information, see <i>Example of Using User Parameters in a Business Process</i>.</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm1 option. However, you can override this parameter using BPML.</p>
User Parameter 2 (userParm2)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm2</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm2 option. However, you can override this parameter using BPML.</p>
User Parameter 3 (userParm3)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm3</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm3 option. However, you can override this parameter using BPML.</p>
User Parameter 4 (userParm4)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm4</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm4 option. However, you can override this parameter using BPML.</p>
User Parameter 5 (userParm5)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm5</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm5 option. However, you can override this parameter using BPML.</p>
Run As User	<p>Applies to the scheduling of the business process. The Run As User field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid value is any valid Application user ID</p> <p>Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit administrative rights (for this run of the business process only), and can enable the scheduled run.</p>

Field	Description
Use 24 Hour Clock Display	If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.
Schedule	<p>Information about scheduling the business process after the File System adapter collects files. The Schedule field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If you select this field, the adapter does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.
Extraction folder (extractionFolder)	<p>The name of the folder or subfolder on the same computer where Application is installed and where it extracts (or writes) data from the primary document as part of a business process. If you do not include the file path for the folder as part of the name, the folder is assumed to be the Application working directory. Required.</p> <p>Note: This parameter is read-only in the GPM.</p>
Unobscure File Contents? (unobscure)	<p>Whether to unobscure the file contents when extracting. Does not work with "exportFile".</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – File contents will be unobscured ◆ No – File contents will not be unobscured <p>Note: This parameter is read-only in the GPM.</p>

Field	Description
Filenaming convention (assignFilename)	Whether to override the document file name and use the assigned file name. Required. Valid values: <ul style="list-style-type: none"> ◆ Use the original file name as the extracted file name – Keeps the names of the files. Note: If the primary document has no document name, the adapter will use a default filename in the form of <i>nodename_yyyyMMddHHmssSSS.dat</i> . <ul style="list-style-type: none"> ◆ Assign a specific name – Gives you the option to navigate to a screen and specify a different filename for the file extracted to the file system. Note: This parameter is read-only in the GPM.
Filename (assignedFilename)	File name you want to assign, including the file name extension. The Filename field only displays if the <i>Filenaming convention</i> is set to Assign a specific name. Required. This field cannot be left blank. You can use “%^” to assign a unique file name in the format <i>nodename_yyyyMMddHHmssSSS</i> . For example, specifying %^.dat as the Filename assigns the name <i>nodename_20040203114020982.dat</i> to the file. Note: This field can also be assigned in the GPM. If you select a filename using the File System adapter configuration, you cannot override it using the GPM assignedFilename parameter. However, you can override it using BPML.
Enable Service for Business Processes	Whether to enable the service for use by business processes. If not selected, the service will be disabled. For more information, see <i>Managing Services and Adapters</i> .

Example of Using User Parameters in a Business Process

The user parameters User Parameter 1 (UserParm1) through User Parameter 5 (UserParm5) in the File System adapter are places to store hard-coded values for use by other services. They are simple assign statements in the BPML.

For example, the Document Extraction service requires the following parameters when used for EDI XML extraction:

```

XMLEDIEnvelopeStandard
XMLRootTag
XMLSenderIDPath
XMLReceiverIDPath
XMLAcceptorLookupAliasPath

```

The File System adapter could be used in an intermediate business process to pass the parameters to the Document Extraction service through the use of User Parameters. The values for the above parameters would be stored in User Parameter 1 through User Parameter 5 in the File System adapter. The following assign statements would then be entered in the BPML code:

```

UserParm1 = XMLEDIEnvelopeStandard
UserParm2 = XMLRootTag

```

UserParm3 = XMLSenderIDPath
 UserParm4 = XMLReceiverIDPath
 UserParm5 = XMLAcceptorLookupAliasPath

The values stored in the User Parameters would be passed through to the Document Extraction service under the assigned parameter names.

Graphical Process Modeler Configuration

The following screen shows the graphical view of the GPM parameters for the File System adapter. The dimmed values have been specified using the File System adapter configuration. The active fields are fields that cannot be configured in the Application or those that are being overridden. There are no fields to be configured on the **Message From Service** tab.

Screen 1 of 3

Service Editor - File System Adapter

Name: File System Adapter

Config: ExampleCollectionFSA

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: FileSystemInputMessage

Name	Value
Action	Collection
appendOnExtract	
assignedFilename	Output_Sample_Data_FulfillOrder.xml
assignFilename	Use the original filename as the extracted filename
attachFile	
bootstrap	No
checkDelete	
collectionFolder	c:\collect
collectMultiple	
collectMultiplePDname	
collectMultiplePrefix	
collectZeroByteFiles	No
dbCollect	

Screen 2 of 3

Service Editor-File System Adapter

Name: File System Adapter

Config: ExampleCollectionFSA

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: FileSystemInputMessage

Name	Value
dbPurgeCollectMin	
deleteAfterCollect	No
docStorageType	System Default
extractionFolder	c:\extract
fileModTimeThreshold	60
filter	*.po
genReport	No
initialWorkFlowId	FIND_MESSAGE_CONSUMER
keepPath	Yes
maxCollect	
maxThreads	10
noFilesSetSuccess	No
obscure	No

Screen 3 of 3

Service Editor-File System Adapter

Name: File System Adapter

Config: ExampleCollectionFSA

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: FileSystemInputMessage

Name	Value
streamBufSize	
subCharsOnExtract	
unobscure	No
userParm1	
userParm2	
userParm3	
userParm4	
userParm5	
useSubFolders	No

The following example shows the corresponding BPML parameters for the File System adapter GPM parameters.

```
<process name="ExampleFileCollection">
  <operation name="File System Adapter">
    <participant name="ExampleCollectionFSA"/>
    <output message="FileSystemInputMessage">
      <assign to="." from="*" />
      <assign to="Action">FS_COLLECT</assign>
      <assign to="collectZeroByteFiles">>false</assign>
      <assign to="deleteAfterCollect">>false</assign>
    </output message>
  </operation>
</process>
```

```

<assign to="fileModTimeThreshold">60</assign>
<assign to="filter">*.po</assign>
<assign to="initialWorkflowId">FIND_MESSAGE_CONSUMER</assign>
<assign to="maxThreads">10</assign>
<assign to="noFilesSetSuccess">>false</assign>
</output>
<input message="inmsg">
<assign to="." from="*" />
</input>
</operation>
</process>

```

The following table describes the fields used to configure the File System adapter in the GPM. This table contains only the fields that are configured in the GPM. The values in parentheses represent the corresponding BPML values. This information is provided for your reference.

Field	Description
Config (participant name)	Name of the adapter configuration. Required. No default.
Action	Action that the File System adapter is to perform. Required. No default. Valid values: <ul style="list-style-type: none"> ◆ Collection (FS_COLLECT) – Files are collected or picked up from the specified folder. ◆ Extraction (FS_EXTRACT) – Files are extracted or written to the specified folder.
appendOnExtract	Whether to append the data if the extract file already exists. Normally, files are overwritten when extracting. This parameter allows you to append the data to the existing files instead. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – Data is appended to existing files. ◆ No (false) – Existing files are overwritten. Default
attachFile	Used to attach a file to a workflow as the primary document. The adapter does not perform any I/O and does not delete the file. Any valid filename is a valid value.
checkDelete	Determines if checking for deletion is possible before collecting files. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – Default ◆ No (false)
collectMultiple	Used to collect multiple files in non-bootstrap mode. Collected files are placed into process data. <ul style="list-style-type: none"> ◆ Yes (true) ◆ No (false) – Default

Field	Description
collectMultiplePDname	Used when collecting multiple files to specify which file will be the primary document. Any valid filename is a valid value.
collectMultiplePrefix	Specifies a prefix to be added to the document name. When multiple documents are created in process data, the documents are named Document1 through DocumentX. Multiple instances could overwrite the documents. You can use this prefix to differentiate the documents in different instances. For example, One instance could use the prefix Inst1_ and another instance could use the prefix Inst2_. The first instance would produce files named Inst1_DocumentX and the second instance would produce files named Inst2_DocumentX. The actual file name is placed as an attribute (filename=) in the document tag. The default value is FSA_.
collectZeroByteFiles (true/false)	Whether to collect zero-byte files. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – Zero-byte files are collected. ◆ No (false) – Zero-byte files are ignored. Default.
concatenateFiles	Used when the collectMultiple option is true and when the File System adapter is set in a non-bootstrap mode. The content of multiple non-zero byte files are concatenated into a single file, and is placed as the primary document. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes (true). Default. ◆ No (false)
dbCollect	If you set this field to true and the field deleteAfterCollect is set to true (which is the default), a database record will be written for every file collected. Before a file is collected, the database is checked to see if the file has already been collected. Optional. Valid values are Yes (true) and No (false). Default is No (false).
dbPurgeCollectMin	Used when dbCollect is set to Yes (true) to specify the number of minutes, from the time the database record is written, before the record is purged. Set this value slightly higher than the scheduled collection interval to prevent duplication before purging. Optional. Valid value is any valid (positive) integer value. Default is 1440 (one day).
deleteAfterCollect	Whether to delete the file after collection. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – File is deleted from the Collection folder after it is collected. Default. ◆ No (false) – File is left in the folder after it is read into Application.
fileModTimeThreshold	Sets the file modification time threshold (in seconds) for files to be collected. A file is collected only if the modification time of the file is older than the number of seconds specified. This prevents premature collection of a file. Defaults to 30 seconds if you do not specify a value.

Field	Description
filter	<p>Collect only files that match a specified filter within the collection folder. Optional. Examples include:</p> <ul style="list-style-type: none"> ◆ *.txt (collects only .txt files) ◆ *.dat (collects only .dat files) ◆ EDI.* (collects only files named EDI with any file extension) ◆ EDI.txt (collect only files named EDI with a file extension of .txt) <p>Note: If there are multiple files in the collection folder and you leave this field blank, one of the following occurs:</p> <ul style="list-style-type: none"> ◆ If the adapter is configured to start a business process, it processes all files placed in the collection folder. ◆ If the adapter is within a business process, it collects only the first file in the collection folder. <p>Note: If you specified this option using the File System adapter configuration, this field will be read-only. However, you can override this parameter using BPML.</p>
genReport	<p>Determines whether a workflow status report is generated for all files regardless of whether they were successfully collected or not. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Status report will be generated whether file collection was successful or unsuccessful. Default ◆ No (false) – Status report will only be generated if file collection is unsuccessful.
maxCollect	Sets the maximum number of files to collect. The default is -1 (unlimited).
maxThreads	Used for performance tuning to set the maximum number of threads used when collecting files. The default is ten threads.
noFilesSetSuccess	<p>Used to determine the workflow status when no files are available to collect in non-bootstrap mode. Optional.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – The workflow status is set to Success even if no files exist in the specified collection folder during collection. ◆ No (false) – The workflow status is set to Error if no files exist in the specified collection folder during collection. <p>The default is No (false)</p>
sortBy	<p>When collectMultiple option is true and the File System Adapter is configured in a non-bootstrap mode, the files are sorted by File name or Modified date as selected in the GPM. The default setting places the files the way they were placed in the Collection folder. Optional.</p> <p>Valid Values: none.</p>
streamBufSize	Used for performance tuning to override the default buffer size of 5k (5120). Optional. Valid value is any integer.

Field	Description
subCharsOnExtract	If the document name contains illegal filename characters, you can use this field to have them replaced with something else. An example would be if the document name was a GUID that contains colons ':' which are illegal in a Windows filename. In this case you would enter ":_ " to replace all occurrences of the colon with an underscore. Optional. Must be entered in two character pairs with no delimiters or spaces. The first character is the one to be replaced, the second is the replacement itself.

Usage Examples

This section contains additional examples using the File System adapter for collecting and extracting files. Examples are included using both the GPM and BPML.

File Collection

The following example using the GPM illustrates a business process that performs a file collection operation when the business process is started.

The screenshot shows the configuration for a File System Adapter service. The configuration name is 'ExampleCollectionFSA'. The message name is 'FileSystemInputMessage'. The output message is 'Obtain Message first, then Process Data'. The configuration table is as follows:

Name	Value
Action	Collection
appendOnExtract	
assignedFilename	Output_Sample_Data_FulfillOrder.xml
assignFilename	Use the original filename as the extracted filename
bootstrap	No
collectionFolder	D:\Test\TestCollectionFolder
collectZeroByteFiles	
deleteAfterCollect	
extractionFolder	D:\SCDocumentationProject\ServicesAdapter\FileSystemAd...
fileModTimeThreshold	
filter	
initialWorkflowId	[Not Applicable]
keepPath	Yes
maxThreads	
noFilesSetSuccess	
useStreams	
useSubFolders	No

Annotations in the image:

- A callout box points to the 'Collection' value in the Action field: "The value Collection for the Action field indicates file collection."
- A callout box points to the 'D:\Test\TestCollectionFolder' value in the collectionFolder field: "The collectionFolder field value was overridden."

The following example illustrates the same business process using BPML.

```

<process name = "ExampleOverridenCollectionFolder">
  <operation name="File System Adapter">
    <participant name="ExampleCollectionFSA"/>
    <output message="FileSystemInputMessage">
      <assign to="Action">FS_COLLECT</assign>
      <assign to="collectionFolder">D:\Test\TestCollectionFolder</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>

```

FS_COLLECT indicates file collection.

Use BPML to override the Collection Folder that was defined in the File System adapter configuration. This method lets you use the same file system adapter configuration to collect files from multiple directories.

File Extraction

The following example using the GPM illustrates a business process that performs a file extraction operation when the business process is started. You could configure a File System adapter to start this business process after files are collected.

ExampleFileExtraction.bp

Service Editor-File System Adapter

Name: File System Adapter

Config: ExampleExtractionFSA

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: FileSystemInputMessage

Action	Name	Value
appendOnExtract		
assignedFilename		
assignFilename		Assign a specific name
bootstrap		No
collectionFolder		D:\SCDocumentationProject\ServicesAdapter\FileSystemAd...
collectZeroByteFiles		
deleteAfterCollect		
extractionFolder		D:\SCDocumentationProject\ServicesAdapter\FileSystemAd...
fileModTimeThreshold		
filter		
initialWorkflowId		[Not Applicable]
keepPath		No
maxThreads		
noFilesSetSuccess		
useStreams		
useSubFolders		Yes

Indicates file extraction

This parameter value was overridden in the File Collection example.

The following example illustrates the same business process using BPML.

```
<process name = "ExampleFileExtraction">
  <operation name="File System Adapter">
    <participant name="FS1"/>
    <output message="FileSystemInputMessage">
      <assign to="Action">FS_EXTRACT</assign>
      <assign to="assignedFilename">OverridenFileName</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

FS_EXTRACT indicates file extraction.

Use BPML to override the assigned filename that was specified in the File System adapter configuration for the extracted file.

File System Adapter (Build 4325 or higher)

The following table provides a high-level overview of the File System adapter:

System name	FileSystem
Graphical Process Modeler (GPM) category	All Services
Description	Collects (imports) files from the file system and extracts (exports) files to the file system. The collected file becomes the primary document in a business process for file collection. A primary document is input to the File System adapter for file extraction.
Business usage	To read files from disk or write files to disk.
Usage example	Commonly used at the beginning of a business process to bootstrap a workflow by reading one or more files from disk and starting a business process. Another common use is to write files to disk for some external purpose. Note: The term <i>bootstrap</i> is used in the Graphical Process Modeler to indicate that the File System adapter is used to start a business process after file collection.
Preconfigured?	No
Requires third party files?	None
Platform availability	All supported platforms for Application
Related services	None
Application requirements	None
Initiates business processes?	Yes, if you define a business process to start when you configure the File System adapter. The business process starts once files are collected.
Invocation	Normally, only the extraction side of the File System adapter is used in a business process, because you can configure the collection side of the File System adapter to start (“bootstrap”) a business process. However, you can include the File System adapter directly in a business process to perform an explicit collection of files.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – File System collection or extraction was successful.◆ Error – File System collection or extraction was unsuccessful.
Restrictions	You must have read access to files and directories for file collection and write access to directories for file extraction.
Persistence level	System Default (Full)

Testing considerations The best (and easiest) way to test this adapter is to set up a business process that only performs a file system extraction and specify that business process as the initial workflow to start (bootstrap).

How the File System Adapter Works

Use the File System adapter to collect (import) files from a file system into a business process and extract (export) files from a business process to a file system. You can configure the File System adapter to start a business process after files are collected from the file system or include the File System adapter in a business process flow. In addition, you can schedule the File System adapter to run at specific time intervals.

You can create multiple File System adapter configurations, one for each of several collection folders. Alternatively, you can use a single File System adapter configuration to point to different directories by specifying the directories for file collection and extraction explicitly in a business process. See *Usage Examples*.

The following sections describe a business scenario in which you could use the File System adapter, along with some sample solutions.

Business Scenario

Your company receives a purchase order from a trading partner in EDI file format and the file is stored on the internal file system. You need to translate the EDI file into XML format and write the translated file to a local directory.

Business Solution Example

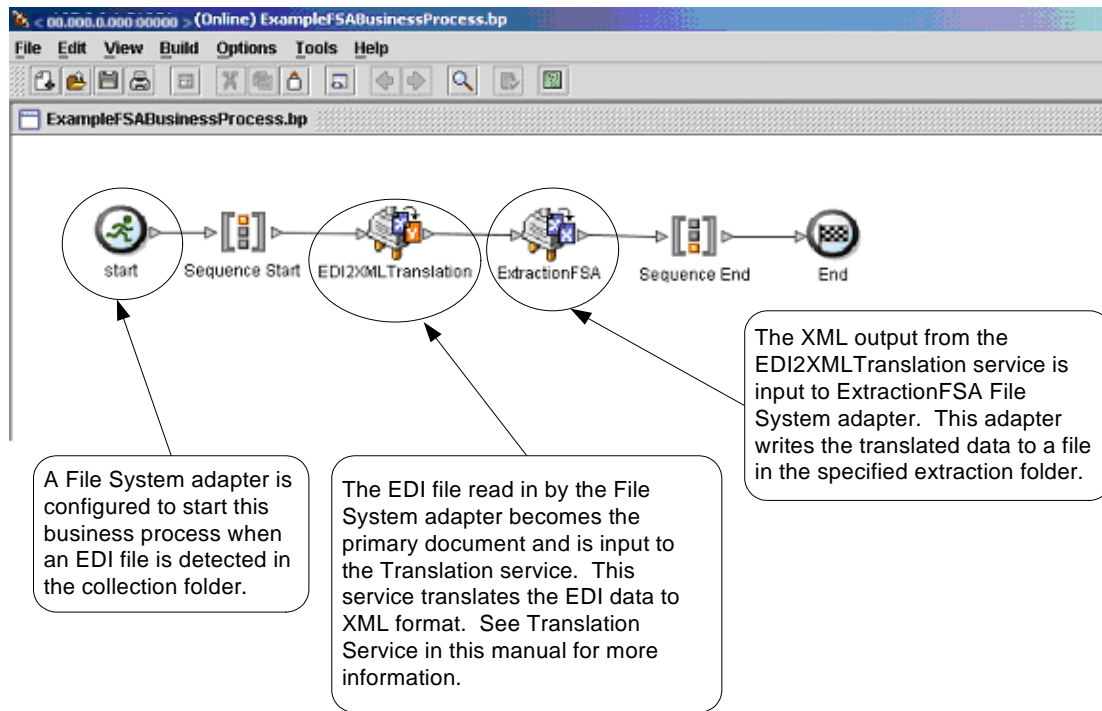
The following approaches are used to solve the above business scenario.

1. Configure a File System adapter instance to be included in a business process to perform a file extraction.
2. Create a business process that translates the EDI file to XML format and then uses the File System adapter instance configured above to extract the resultant XML data to the file system
3. Configure a separate File System adapter instance to start the business process created in the previous step after an EDI file is detected in the collection folder. This File System adapter instance is also scheduled to run at 30-minute intervals.

This business solution is described for both the Graphical Process Modeler (GPM) and the Business Process Modeling Language (BPML).

Graphical Process Modeler (GPM) Example

The following example shows a simple solution to the above business scenario using the GPM.



Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML.

```
<process name="ExampleFSABusinessProcess">
  <sequence name="Sequence Start">
    <operation name="EDI2XMLTranslation">
      <participant name="Translation"/>
      <output message="TranslationTypeInputMessage">
        <assign to="exhaust_input">YES</assign>
        <assign to="map_name">ExampleP0850</assign>
        <assign to="output_to_process_data">NO</assign>
        <assign to="validate_input">NO</assign>
        <assign to="validate_input_against_dtd">NO</assign>
        <assign to="validate_output">NO</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="ExtractionFSA">
      <participant name="ExampleExtractionFSA"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```



Implementing the File System Adapter

You can implement the File System adapter in three ways:

- Collect files within a business process.
- Extract files to the file system from a business process.
- Collect files and then start a new business process.

The information in this section applies to all three implementations.

Before you begin to implement a File System adapter, you need to collect the following information:

- The name of the business process (if the adapter is to start a business process)
- The directory path from which files are collected
- The directory path to which files are extracted

Process Overview

To implement the File System adapter, complete the following tasks:

1. Create a File System adapter configuration.
2. Configure the File System adapter.

3. Create a business process to run after the File System adapter collects files, or create and enable a business process that includes the File System adapter (collecting or extracting files).
4. Test the business process and the adapter.
5. Run the business process.

Configuring the File System Adapter

To create a File System adapter configuration, specify field settings in Application and in the GPM.

File System Configuration


The following table describes the fields used to configure the File System adapter in the Application.

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a group	Group to associate with the adapter. Valid values: <ul style="list-style-type: none"> ◆ None – No group is selected ◆ Create New Group – Allows the creation of a new group ◆ Select Group – Select from a list of available groups
Collection folder (collectionFolder)	The name of the folder or subfolder on the same computer where Application is installed and where it collects (or picks up) files as part of a business process. If the path for the folder is not included as part of the name, the folder is assumed to be in the Application working directory. Required. Note: <ul style="list-style-type: none"> ◆ The deleteAfterCollect parameter in the GPM defaults to Yes. If you do not change the default value to No, files that are collected are deleted from the Collection Folder. The File System adapter does not copy the files it collects for processing. See <i>Graphical Process Modeler Configuration</i> for information about the deleteAfterCollect parameter. ◆ The collectionFolder parameter is read-only in the GPM. However, you can override this parameter using BPML.

Field	Description
Filename filter (filter)	<p>Collect only files that match a specified filter within the collection folder. Optional.</p> <p>Examples include:</p> <ul style="list-style-type: none"> ◆ *.txt (collects only .txt files) ◆ *.dat (collects only .dat files) ◆ EDI.* (collects only files named EDI with any file extension) ◆ EDI.txt (collect only files named EDI with a file extension of .txt) <p>Note: If there are multiple files in the collection folder and you leave this field blank, one of the following occurs:</p> <ul style="list-style-type: none"> ◆ If the adapter is configured to start a business process, it processes all files placed in the collection folder. ◆ If the adapter is within a business process, it collects only the first file in the collection folder. <p>Note: If you specify this option using the File System adapter configuration, you cannot override the value using the GPM filter parameter. However, you can override this parameter using BPML.</p>
Collect files from subfolders within and including the collection folder? (useSubFolders)	<p>Whether to scan for files in subfolders of the collection folder. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Collects files in the specified folder and all subfolders. ◆ No – Collects files in the specified folder only. <p>Note: This parameter is read-only in the GPM.</p>
Use the absolute file path name for the document name? (keepPath)	<p>Whether to keep the absolute path name of the files collected when assigning the document name. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – The absolute file path name is kept with the document in the business process. Choose this value if your business process requires the path information to precede the file name. ◆ No – Only the file name is kept with the document in the business process. <p>Note: An absolute path is a path that points to the same location regardless of the working directory or combined paths. It is usually written in reference to a root directory. For example, c:\dir1\subdir1\somefile.txt (Windows) and /home/dir1/subdir1/somefile.txt (UNIX) are examples of absolute paths to the file somefile.txt.</p> <p>Note: This parameter is read-only in the GPM.</p>

Field	Description
Start a business process once files are collected? (bootstrap)	<p>Whether to start a business process using the File System adapter after files are collected. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Starts the business process specified from the business process drop-down list. <p>Note: An instance of the business process is started for every file that matches the filtering criteria specified for file collection until the number of threads specified on the maxThreads parameter is reached. See <i>Graphical Process Modeler Configuration</i> for information about the maxThreads parameter.</p> <ul style="list-style-type: none"> ◆ No – No business process will be started. <p>Note: This parameter is read-only in the GPM.</p>
Business Process (initialWorkflowId)	<p>The business process to start after files are collected. Required when <i>Start a business process</i> is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Name of the business process to start ◆ Not Applicable <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify a business process using the configuration, you cannot override this value using the GPM initialWorkflowId option. If you select Not Applicable, a business process can be selected in the GPM. In either case, you can override this parameter using BPML.</p>
Document storage type (docStorageType)	<p>Defines how the document will be stored in the system. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM DocStorageType option. However, you can override this parameter using BPML.</p> <p>Note: For more information about document storage types, see <i>Selecting a Document Storage Method for Bootstrap Adapters</i> on page 20.</p>
Obscure File Contents? (obscure)	<p>Specifies whether to obscure the file contents when collecting. Does not work with “attachFile” or “importFile”.</p> <ul style="list-style-type: none"> ◆ Yes – File contents will be obscured ◆ No – File contents will not be obscured <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM Obscure option. However, you can override this parameter using BPML.</p>

Field	Description
User Parameter 1 (userParm1)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm1. For more information, see <i>Example of Using User Parameters in a Business Process</i>.</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm1 option. However, you can override this parameter using BPML.</p>
User Parameter 2 (userParm2)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm2</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm2 option. However, you can override this parameter using BPML.</p>
User Parameter 3 (userParm3)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm3</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm3 option. However, you can override this parameter using BPML.</p>
User Parameter 4 (userParm4)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm4</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm4 option. However, you can override this parameter using BPML.</p>
User Parameter 5 (userParm5)	<p>A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm5</p> <p>Note: This field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm5 option. However, you can override this parameter using BPML.</p>
Run As User	<p>Applies to the scheduling of the business process. The Run As User field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid value is any valid Application user ID</p> <p>Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit administrative rights (for this run of the business process only), and can enable the scheduled run.</p>

Field	Description
Use 24 Hour Clock Display	If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.
Schedule	<p>Information about scheduling the business process after the File System adapter collects files. The Schedule field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If you select this field, the adapter does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.
Extraction folder (extractionFolder)	<p>The name of the folder or subfolder on the same computer where Application is installed and where it extracts (or writes) data from the primary document as part of a business process. If you do not include the file path for the folder as part of the name, the folder is assumed to be the Application working directory. Required.</p> <p>Note: This parameter is read-only in the GPM.</p>
Unobscure File Contents? (unobscure)	<p>Whether to unobscure the file contents when extracting. Does not work with "exportFile".</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – File contents will be unobscured ◆ No – File contents will not be unobscured <p>Note: This parameter is read-only in the GPM.</p>

Field	Description
Filenaming convention (assignFilename)	Whether to override the document file name and use the assigned file name. Required. Valid values: <ul style="list-style-type: none"> ◆ Use the original file name as the extracted file name – Keeps the names of the files. Note: If the primary document has no document name, the adapter will use a default filename in the form of <i>nodename_yyyyMMddHHmmssSSS.dat</i> . <ul style="list-style-type: none"> ◆ Assign a specific name – Gives you the option to navigate to a screen and specify a different filename for the file extracted to the file system. Note: This parameter is read-only in the GPM.
Filename (assignedFilename)	File name you want to assign, including the file name extension. The Filename field only displays if the <i>Filenaming convention</i> is set to Assign a specific name. Required. This field cannot be left blank. You can use “%^” to assign a unique file name in the format <i>nodename_yyyyMMddHHmmssSSS</i> . For example, specifying %^.dat as the Filename assigns the name <i>nodename_20040203114020982.dat</i> to the file. Note: This field can also be assigned in the GPM. If you select a filename using the File System adapter configuration, you cannot override it using the GPM assignedFilename parameter.
Enable Service for Business Processes	Whether to enable the service for use by business processes. If not selected, the service will be disabled. For more information, see <i>Managing Services and Adapters</i> .

Example of Using User Parameters in a Business Process

The user parameters User Parameter 1 (UserParm1) through User Parameter 5 (UserParm5) in the File System adapter are places to store hard-coded values for use by other services. They are simple assign statements in the BPML.

For example, the Document Extraction service requires the following parameters when used for EDI XML extraction:

```

XMLEDIEnvelopeStandard
XMLRootTag
XMLSenderIDPath
XMLReceiverIDPath
XMLAcceptorLookupAliasPath

```

The File System adapter could be used in an intermediate business process to pass the parameters to the Document Extraction service through the use of User Parameters. The values for the above parameters would be stored in User Parameter 1 through User Parameter 5 in the File System adapter. The following assign statements would then be entered in the BPML code:

```

UserParm1 = XMLEDIEnvelopeStandard
UserParm2 = XMLRootTag
UserParm3 = XMLSenderIDPath

```

UserParm4 = XMLReceiverIDPath

UserParm5 = XMLAcceptorLookupAliasPath

The values stored in the User Parameters would be passed through to the Document Extraction service under the assigned parameter names.

Graphical Process Modeler Configuration

The following screen shows the graphical view of the GPM parameters for the File System adapter. The dimmed values have been specified using the File System adapter configuration. The active fields are fields that cannot be configured in the Application or those that are being overridden. There are no fields to be configured on the **Message From Service** tab.

Screen 1 of 3

Name	Value
Action	Collection
appendOnExtract	
assignedFilename	Output_Sample_Data_FulfillOrder.xml
assignFilename	Use the original filename as the extracted filename
attachFile	
bootstrap	No
checkDelete	
collectionFolder	c:\collect
collectMultiple	
collectMultiplePDname	
collectMultiplePrefix	
collectZeroByteFiles	No
dbCollect	

Screen 2 of 3

Service Editor-File System Adapter

Name: File System Adapter

Config: ExampleCollectionFSA

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: FileSystemInputMessage

Name	Value
dbPurgeCollectMin	
deleteAfterCollect	No
docStorageType	System Default
extractionFolder	c:\extract
fileModTimeThreshold	60
filter	*.po
genReport	No
initialWorkFlowId	FIND_MESSAGE_CONSUMER
keepPath	Yes
maxCollect	
maxThreads	10
noFilesSetSuccess	No
obscure	No

Screen 3 of 3

Service Editor-File System Adapter

Name: File System Adapter

Config: ExampleCollectionFSA

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: FileSystemInputMessage

Name	Value
streamBufSize	
subCharsOnExtract	
unobscure	No
userParm1	
userParm2	
userParm3	
userParm4	
userParm5	
useSubFolders	No

The following example shows the corresponding BPML parameters for the File System adapter GPM parameters.

```
<process name="ExampleFileCollection">
  <operation name="File System Adapter">
    <participant name="ExampleCollectionFSA"/>
    <output message="FileSystemInputMessage">
      <assign to="." from="*" />
      <assign to="Action">FS_COLLECT</assign>
      <assign to="collectZeroByteFiles">>false</assign>
      <assign to="deleteAfterCollect">>false</assign>
    </output message>
  </operation>
</process>
```



```

<assign to="fileModTimeThreshold">60</assign>
<assign to="filter">*.po</assign>
<assign to="initialWorkflowId">FIND_MESSAGE_CONSUMER</assign>
<assign to="maxThreads">10</assign>
<assign to="noFilesSetSuccess">>false</assign>
</output>
<input message="inmsg">
<assign to="." from="*" />
</input>
</operation>
</process>

```

The following table describes the fields used to configure the File System adapter in the GPM. This table contains only the fields that are configured in the GPM. The values in parentheses represent the corresponding BPML values. This information is provided for your reference.

Field	Description
Config (participant name)	Name of the adapter configuration. Required. No default.
Action	Action that the File System adapter is to perform. Required. No default. Valid values: <ul style="list-style-type: none"> ◆ Collection (FS_COLLECT) – Files are collected or picked up from the specified folder. ◆ Extraction (FS_EXTRACT) – Files are extracted or written to the specified folder.
appendOnExtract	Whether to append the data if the extract file already exists. Normally, files are overwritten when extracting. This parameter allows you to append the data to the existing files instead. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – Data is appended to existing files. ◆ No (false) – Existing files are overwritten. Default
attachFile	Used to attach a file to a workflow as the primary document. The adapter does not perform any I/O and does not delete the file. Any valid filename is a valid value.
checkDelete	Determines if checking for deletion is possible before collecting files. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – Default ◆ No (false)
collectMultiple	Used to collect multiple files in non-bootstrap mode. Collected files are placed into process data. <ul style="list-style-type: none"> ◆ Yes (true) ◆ No (false) – Default

Field	Description
collectMultiplePDname	Used when collecting multiple files to specify which file will be the primary document. Any valid filename is a valid value.
collectMultiplePrefix	Specifies a prefix to be added to the document name. When multiple documents are created in process data, the documents are named Document1 through DocumentX. Multiple instances could overwrite the documents. You can use this prefix to differentiate the documents in different instances. For example, One instance could use the prefix Inst1_ and another instance could use the prefix Inst2_. The first instance would produce files named Inst1_DocumentX and the second instance would produce files named Inst2_DocumentX. The actual file name is placed as an attribute (filename=) in the document tag. The default value is FSA_.
collectZeroByteFiles (true/false)	Whether to collect zero-byte files. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – Zero-byte files are collected. ◆ No (false) – Zero-byte files are ignored. Default.
concatenateFiles	Used when the collectMultiple option is true and when the File System adapter is set in a non-bootstrap mode. The content of multiple non-zero byte files are concatenated into a single file, and is placed as the primary document. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes (true). Default. ◆ No (false)
dbCollect	If you set this field to true and the field deleteAfterCollect is set to true (which is the default), a database record will be written for every file collected. Before a file is collected, the database is checked to see if the file has already been collected. Optional. Valid values are Yes (true) and No (false). Default is No (false).
dbPurgeCollectMin	Used when dbCollect is set to Yes (true) to specify the number of minutes, from the time the database record is written, before the record is purged. Set this value slightly higher than the scheduled collection interval to prevent duplication before purging. Optional. Valid value is any valid (positive) integer value. Default is 1440 (one day).
deleteAfterCollect	Whether to delete the file after collection. Valid values: <ul style="list-style-type: none"> ◆ Yes (true) – File is deleted from the Collection folder after it is collected. Default. ◆ No (false) – File is left in the folder after it is read into Application.
fileModTimeThreshold	Sets the file modification time threshold (in seconds) for files to be collected. A file is collected only if the modification time of the file is older than the number of seconds specified. This prevents premature collection of a file. Defaults to 30 seconds if you do not specify a value.

Field	Description
filter	<p>Collect only files that match a specified filter within the collection folder. Optional. Examples include:</p> <ul style="list-style-type: none"> ◆ *.txt (collects only .txt files) ◆ *.dat (collects only .dat files) ◆ EDI.* (collects only files named EDI with any file extension) ◆ EDI.txt (collect only files named EDI with a file extension of .txt) <p>Note: If there are multiple files in the collection folder and you leave this field blank, one of the following occurs:</p> <ul style="list-style-type: none"> ◆ If the adapter is configured to start a business process, it processes all files placed in the collection folder. ◆ If the adapter is within a business process, it collects only the first file in the collection folder. <p>Note: If you specified this option using the File System adapter configuration, this field will be read-only. However, you can override this parameter using BPML.</p>
genReport	<p>Determines whether a workflow status report is generated for all files regardless of whether they were successfully collected or not. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – Status report will be generated whether file collection was successful or unsuccessful. Default ◆ No (false) – Status report will only be generated if file collection is unsuccessful.
maxCollect	<p>Sets the maximum number of files to collect. The default is -1 (unlimited).</p>
maxThreads	<p>Used for performance tuning to set the maximum number of threads used when collecting files. The default is ten threads.</p>
noFilesSetSuccess	<p>Used to determine the workflow status when no files are available to collect in non-bootstrap mode. Optional.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes (true) – The workflow status is set to Success even if no files exist in the specified collection folder during collection. ◆ No (false) – The workflow status is set to Error if no files exist in the specified collection folder during collection. <p>The default is No (false)</p>
sortBy	<p>When collectMultiple option is true and the File System Adapter is configured in a non-bootstrap mode, the files are sorted by File name or Modified date as selected in the GPM. The default setting places the files the way they were placed in the Collection folder. Optional.</p> <p>Valid Values: none.</p>
streamBufSize	<p>Used for performance tuning to override the default buffer size of 5k (5120). Optional. Valid value is any integer.</p>

Field	Description
subCharsOnExtract	If the document name contains illegal filename characters, you can use this field to have them replaced with something else. An example would be if the document name was a GUID that contains colons ':' which are illegal in a Windows filename. In this case you would enter ":_ " to replace all occurrences of the colon with an underscore. Optional. Must be entered in two character pairs with no delimiters or spaces. The first character is the one to be replaced, the second is the replacement itself.

Usage Examples

This section contains additional examples using the File System adapter for collecting and extracting files. Examples are included using both the GPM and BPML.

File Collection

The following example using the GPM illustrates a business process that performs a file collection operation when the business process is started.

The screenshot shows the Service Editor for the File System Adapter. The configuration is set to 'ExampleCollectionFSA'. The 'Message To Service' and 'Message From Service' tabs are selected. The 'Output Msg' is 'Obtain Message first, then Process Data' and the 'Message Name' is 'FileSystemInputMessage'.

Name	Value
Action	Collection
appendOnExtract	
assignedFilename	Output_Sample_Data_FulfillOrder.xml
assignFilename	Use the original filename as the extracted filename
bootstrap	No
collectionFolder	D:\Test\TestCollectionFolder
collectZeroByteFiles	
deleteAfterCollect	
extractionFolder	D:\SCDocumentationProject\ServicesAdapter\FileSystemAd...
fileModTimeThreshold	
filter	
initialWorkflowId	[Not Applicable]
keepPath	Yes
maxThreads	
noFilesSetSuccess	
useStreams	
useSubFolders	No

Annotations in the image:

- A callout box points to the 'Action' field value 'Collection' with the text: "The value Collection for the Action field indicates file collection."
- A callout box points to the 'collectionFolder' field value 'D:\Test\TestCollectionFolder' with the text: "The collectionFolder field value was overridden."

The following example illustrates the same business process using BPML.

```

<process name = "ExampleOverridenCollectionFolder">
  <operation name="File System Adapter">
    <participant name="ExampleCollectionFSA"/>
    <output message="FileSystemInputMessage">
      <assign to="Action">FS_COLLECT</assign>
      <assign to="collectionFolder">D:\Test\TestCollectionFolder</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>

```

FS_COLLECT indicates file collection.

Use BPML to override the Collection Folder that was defined in the File System adapter configuration. This method lets you use the same file system adapter configuration to collect files from multiple directories.

File Extraction

The following example using the GPM illustrates a business process that performs a file extraction operation when the business process is started. You could configure a File System adapter to start this business process after files are collected.

ExampleFileExtraction.bp

Service Editor-File System Adapter

Name: File System Adapter

Config: ExampleExtractionFSA

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: FileSystemInputMessage

Action	Name	Value
appendOnExtract		
assignedFilename		
assignFilename		Assign a specific name
bootstrap		No
collectionFolder		D:\SCDocumentationProject\ServicesAdapter\FileSystemAd...
collectZeroByteFiles		
deleteAfterCollect		
extractionFolder		D:\SCDocumentationProject\ServicesAdapter\FileSystemAd...
fileModTimeThreshold		
filter		
initialWorkflowId		[Not Applicable]
keepPath		No
maxThreads		
noFilesSetSuccess		
useStreams		
useSubFolders		Yes

Indicates file extraction

This parameter value was overridden in the File Collection example.

The following example illustrates the same business process using BPML.

```
<process name = "ExampleFileExtraction">
  <operation name="File System Adapter">
    <participant name="FS1"/>
    <output message="FileSystemInputMessage">
      <assign to="Action">FS_EXTRACT</assign>
      <assign to="assignedFilename">OverridenFileName</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

FS_EXTRACT indicates file extraction.

Use BPML to override the assigned filename that was specified in the File System adapter configuration for the extracted file.

For Each Document Service

The For Each Document service incrementally processes a set of documents. Each time the service runs, it makes the next document in the document set the primary document. This enables business processes to contain loops that operate on each document in turn. The following table provides an overview of the For Each Document service:

System name	ForEachDocument
Graphical Process Modeler (GPM) category	All Services
Description	The For Each Document service incrementally processes a set of documents. Each time runs, it makes the next document in the set the primary document. This enables business processes to contain loops that operate on each document in turn.
Business usage	To loop through multiple documents within a business process.
Usage example	Used when there are multiple documents within a business process that need to be processed. This service is used within the X12 Develope and EDIFACT Develope business processes to iterate document deenvolving over functional groups and transaction sets.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	The Document Extraction service is often used in conjunction with the For Each Document service to extract individual documents out of a batch file.
Application requirements	None
Initiates business processes?	None
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success–The service has set the primary document appropriately.◆ Error–The service encountered an unexpected condition that caused the iteration to fail.
Testing considerations	Create and test a business process that uses this service to incrementally process multiple documents.

How the For Each Document Service Works

Use the For Each Document service when you have multiple documents within the business process that all need to be processed.

The following sections describe a business scenario and sample solutions using the For Each Document service.

Business Scenario

Your company has multiple records that are in a flat file format. You need to extract these records and process each one individually.

Business Solution Example

The following approaches are used to solve the above business scenario.

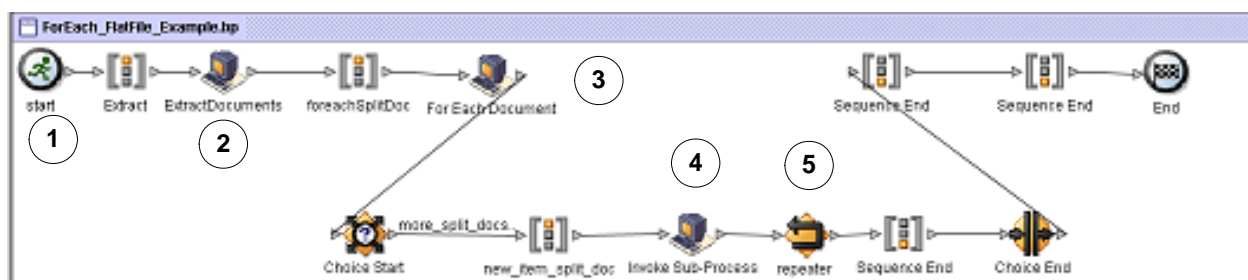
1. Create maps required by the Document Extraction service to extract all the individual documents and place them in the business process.
2. Create a business process that:
 - ◆ Extracts the documents from the flat file using the Document Extraction service.
 - ◆ Uses the For Each Document service to iterate through all of the extracted documents.
 - ◆ Uses a Rule (more_split_docs) created through the Rule Manager and applied through the Edge Editor that checks if all documents have been processed.
 - ◆ Invokes a subprocess that processes each document.

This business solution is described for both the GPM and the Business Process Modeling Language (BPML).

Note: The focus in this example is on how to use the For Each Document service. The details on the subprocess are not included.

GPM Example

The following example shows a solution to the above business scenario using the GPM:



1. A flat file containing multiple records is passed to the business process as the primary document.
2. The Document Extraction service extracts all of the documents from the flat file and places them as individual documents in the process data of the business process.

3. The For Each Document service iterates through the documents, making each successive document the primary document.
4. A subprocess runs to process each of the documents.
5. The Rule (more_split_docs) is applied using the Edge Editor.

The following example shows the GPM configuration of the For Each Document service.

Service Editor-For Each Document

Name For Each Document

Config. ForEachDocument

Message To Service **Message From Service**

Output Msg Obtain Message first, then Process Data

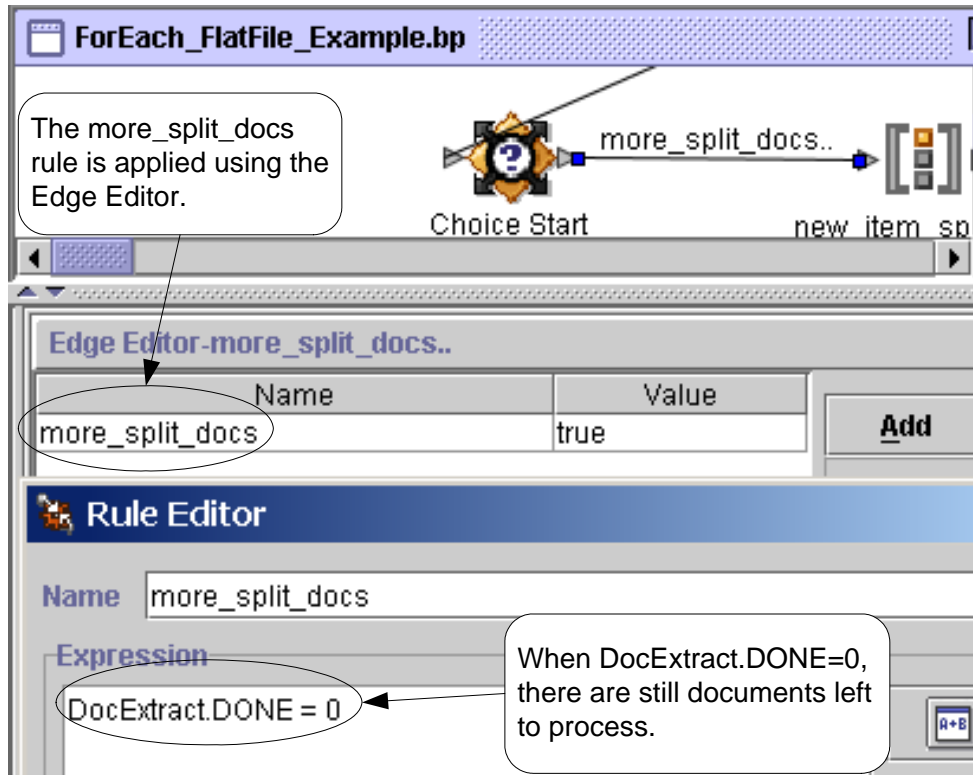
Message Name ForEachDocumentTypeInputMessage

Name	Value	Use XPATH?
DOCUMENT_NAME_PREFIX	DOC-SPLIT-	<input type="checkbox"/>
ITERATOR_NAME	DocExtract	<input type="checkbox"/>

DOC-SPLIT- is the prefix for all of the extracted documents.

DocExtract is the ITERATOR-NAME.

The following graphic shows the Rule Editor and Edge Editor configuration in the GPM for the business process.



DocExtract.DONE is an output parameter from the For Each Document service.

The Rule (more_split_docs) is DocExtract.DONE = 0. This Rule is created using the Rule Manager and applied in the business process using the Edge Editor.

When this rule is true, processing continues. After all the documents are processed, DocExtract.DONE is set to 1 by the For Each Document service and processing stops.

Business Process Modeling (BPML) Example

The following example shows the corresponding business process solution using BPML.

```
<process name="ForEach_FlatFile_Example">
  <rule name="more_split_docs">
    <condition>DocExtract.DONE = 0</condition>
  </rule>
  <sequence name="Extract">
    <operation name="ExtractDocuments">
      <participant name="DocumentExtractionService"/>
      <output message="DocumentExtractionTypeInputMessage">
        <assign to="BatchLikeDocuments">NO</assign>
        <assign to="DocExtractMapList">DocExtract_CTN DocExtract_ITM</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">

```

```

        <assign to="." from="*"></assign>
    </input>
</operation>
<sequence name="foreachSplitDoc">
    <operation name="For Each Document">
        <participant name="ForEachDocument"/>
        <output message="ForEachDocumentTypeInputMessage">
            <assign to="DOCUMENT_NAME_PREFIX">DOC-SPLIT-</assign>
            <assign to="ITERATOR_NAME">DocExtract</assign>
            <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
            <assign to="." from="*"></assign>
        </input>
    </operation>
    <choice name="Choice Start">
        <select>
            <case ref="more_split_docs" activity="new_item_split_doc"/>
        </select>
    <sequence name="new_item_split_doc">
        <operation name="Invoke Sub-Process">
            <participant name="InvokeSubProcessService"/>
            <output message="InvokeSubProcessServiceTypeInputMessage">
                <assign to="INVOKE_MODE">ASYN</assign>
                <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
                <assign to="WFD_NAME"> ProcessNewItemRequest</assign>
                <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
                <assign to="." from="*"></assign>
            </input>
        </operation>
        <repeat name="repeater" ref="foreachSplitDoc"/>
    </sequence>
    </choice>
</sequence>
</sequence>
</process>

```

Process Data

The following shows process data after the documents are extracted and the first document has been processed.

```

<ProcessData>
  <DOC-SPLIT-1 SCIObjectID="L2000-000248:f96c:fb591069a2:-60fb">
    <SenderID>03011</SenderID>
    <ReceiverID/>
    <AcceptorLookupAlias/>
  </DOC-SPLIT-1>
  <DOC-SPLIT-2 SCIObjectID="L2000-000248:f96c:fb591069a2:-60fa">
    <SenderID>03012</SenderID>
    <ReceiverID/>
    <AcceptorLookupAlias/>
  </DOC-SPLIT-2>
  <DOC-SPLIT-3 SCIObjectID="L2000-000248:f96c:fb591069a2:-60f9">
    <SenderID>03014</SenderID>
    <ReceiverID/>
    <AcceptorLookupAlias/>
  </DOC-SPLIT-3>
  <DocExtract SCIObjectID="L2000-000248:f96c:fb591069a2:-60ea"/>
  <CONTRACT_FOUND>NO</CONTRACT_FOUND>
  <PrimaryDocument SCIObjectID="L2000-000248:f96c:fb591069a2:-60e9"/>
  <DocExtract.NAME>DOC-SPLIT-1</DocExtract.NAME>
  <DocExtract.INDEX SCIObjectID="L2000-000248:f96c:fb591069a2:-60e8"/>
  <DocExtract.DONE>0</DocExtract.DONE>
</ProcessData>

```

The following shows the process data after the business process has completed.

```

<ProcessData>
  .
  .
  .
  <CONTRACT_FOUND>NO</CONTRACT_FOUND>
  <DocExtract.DONE>1</DocExtract.DONE>
  <INVOKE_ID_LIST>46396</INVOKE_ID_LIST>
  <INVOKE_ID_LIST>46397</INVOKE_ID_LIST>
  <PrimaryDocument SCIObjectID="L2000-000248:f96c:fb591069a2:-60a1"/>
  <INVOKE_ID_LIST>46398</INVOKE_ID_LIST>
</ProcessData>

```

Note: When the Document Extraction service parameter PDToProcessData is set to No, process data will only contain the current DOC_SPLIT section. Each iteration through the For Each Document service will update process data with the current DOC_SPLIT and remove the previous split.

Output from Service to Business Process

The following table describes the output from the For Each Document service to the business process. The parameters in the table, along with their values, are put into process data for use later in a business process.

Note: In the following table, I is replaced with the value of the ITERATOR_NAME parameter. If the ITERATOR_NAME were TEST, these parameters would be TEST, TEST.INDEX, TEST.NAME, and TEST.DONE.

Parameter	Description
I	ArrayList containing the document data in the order that the documents are processed.
I.INDEX	The index of the document that was most recently made the primary document.
I.NAME	Name of the document that was most recently made the primary document.
I.DONE	Valid values: <ul style="list-style-type: none"> ◆ 0 – There are documents remaining to be processed. ◆ 1 – The iteration is complete.

The following example shows the process data with the output parameters. The array list in this case is named DocExtract.

```
<ProcessData>
.
.
.
<DocExtract SCIObjectID="L2000-000248:f96c:fb591069a2:-60ea" />
<DocExtract.NAME>DOC-SPLIT-1</DocExtract.NAME>
<DocExtract.INDEX SCIObjectID="L2000-000248:f96c:fb591069a2:-60e8" />
<DocExtract.DONE>0</DocExtract.DONE>
.
.
</ProcessData>
```

Usage Examples

The following business processes are predefined in Application:

- The X12Deenvelope business process
- The EDIFACTDeenvelope business process

Implementing the For Each Document Service

To implement the For Each Document service, complete the following tasks:

1. If a configuration does not already exist, create a For Each Document service configuration. For information, see *Managing Services and Adapters*.

Note: A For Each Document service is configured with the installation of Application.
2. Configure the For Each Document service. For information, see *Configuring the For Each Document Service*.
3. Use the For Each Document service in a business process.

Configuring the For Each Document Service

To configure the For Each Document service, you must specify field settings in Application and in the GPM. The following example shows the GPM parameters for the For Each Document service. There are no fields to be configured on the Message From Service tab.

Name	Value	Use XPATH?
DOCUMENT_NAME_PREFIX	DocumentPrefix	<input type="checkbox"/>
ITERATOR_NAME	ExampleIterator	<input type="checkbox"/>

The following example shows the corresponding BPML parameters for the For Each Document service GPM parameters.

```
<operation name="For Each Document">
  <participant name="ForEachDocument" />
  <output message="ForEachDocumentTypeInputMessage">
    <assign to="DOCUMENT_NAME_PREFIX">DocumentPrefix</assign>
    <assign to="ITERATOR_NAME">ExampleIterator</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

GPM Configuration

The following table describes the fields used to configure the For Each Document service in the GPM:

Field	Description
Config (participant name)	Name of the service configuration.
ITERATOR_NAME	Name used to identify this document iterator in the business process. This is the prefix for the tags that are generated by the service and is incremented as it loops through the documents.

Field	Description
DOCUMENT_NAME_PREFIX	<p>Filter that restricts the documents included in the iteration. The filter is the prefix on each document name. When this parameter is used, only documents beginning with this prefix are processed.</p> <p>If DOCUMENT_KEY_PREFIX is used, this parameter is ignored.</p> <p>If neither this parameter or DOCUMENT_KEY_PREFIX is specified, all documents in the business process are processed by the For Each service.</p>
DOCUMENT_KEY_PREFIX	<p>Filter that restricts the documents included in the iteration. The filter is the prefix on the key in process data for each document. When this parameter is used, only documents that have keys beginning with this prefix are processed.</p> <p>The parameter DOCUMENT_NAME_PREFIX is ignored if this parameter is specified.</p> <p>If neither this parameter or DOCUMENT_NAME_PREFIX is specified, all documents in the business process are processed by the For Each service.</p>

FTP Client Adapter

The following table provides an overview of the FTP Client adapter:

System name	FTP Client Adapter
Graphical Process Modeler (GPM) category	Not available in the GPM
Description	<p>This adapter is used in conjunction with the FTP Client services to send FTP requests to trading partners using a Perimeter server. The FTP Client adapter replaces the B2B FTP Client adapter, the FTP Send adapter, and the FTP GET adapter, which are being retired. The FTP Client adapter and its related services provide all the functionality of the former three adapters plus the following enhancements:</p> <ul style="list-style-type: none">◆ Uses Perimeter Services◆ Scriptable through BPML◆ Highly scalable◆ Supports large files (up to 15 GB)◆ Designed to work easily with nearly any FTP server
Business usage	Use this adapter to send FTP requests to perform activities such as to <i>get</i> or <i>put</i> files into a directory on the trading partner's FTP server.
Usage example	A Application business process is executed that translates a document that must be sent to a trading partner. After the translation, Application looks up information about how to transport data to the trading partner in the trading profile. The trading profile specifies FTP as the transport protocol. Application then uses the FTP Client adapter to send the document to the trading partner.
Preconfigured?	No
Requires third party files?	Certicom SSL Library (currently available in Application)
Platform availability	All supported Application platforms

Related services	<p>The FTP Client adapter works with the following services:</p> <ul style="list-style-type: none"> ◆ FTP Client Begin Session service ◆ FTP Client CD service ◆ FTP Client DELETE service ◆ FTP Client End Session service ◆ FTP Client GET service ◆ FTP Client LIST service ◆ FTP Client MOVE service ◆ FTP Client PUT service ◆ FTP Client PWD service ◆ FTP Client QUOTE service ◆ FTP Client SITE service
Application requirements	<p>An FTP server at the external trading partner location.</p> <p>When this adapter is configured with a 'non-local-mode' Perimeter server, the Perimeter server must be installed and running.</p> <p>Note: You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring.</p>
Initiates business processes?	This adapter does not initiate business processes.
Invocation	This adapter is invoked from one of the FTP Client services used within a business process.
Business process context considerations	Business processes using FTP client services should not be marked as Auto Resume. These services require an established session which will no longer exist after a restart.
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	None
Persistence level	None

Testing considerations	<p>Test this adapter by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test the FTP Client adapter, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code> 4. Verify that the business process runs successfully <p>Debug information for this adapter can be found in the FTP Client adapter and services log files.</p>
------------------------	---

Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.
-------	---

Business Scenario

You want to send a file to a trading partner, using FTP. The file must be placed in a specific directory on the trading partner's FTP server. After putting the file in the directory, you want to list the contents of the directory to verify that it was copied correctly.

Business Solution Example

The following approaches are used to solve the above business scenario.

1. Create an FTP Client adapter configuration to be used by the FTP client services that will be included in a business process to send the file.
2. Create a business process that:

- ◆ Opens an FTP communications session with the trading partner (using the FTP Client Begin Session service)
- ◆ Changes to the correct directory on the trading partner's FTP server (using the FTP Client CD service)
- ◆ Puts the file in the directory (using the FTP Client PUT service)
- ◆ Generates a list of the directory contents to verify that the put command was successful (using the FTP Client LIST service)
- ◆ Closes the FTP session (using the FTP Client End Session service)

This business solution is described for both the GPM and the BPML.

GPM Example

The following example shows a solution to the above business scenario using the GPM:

The screenshot displays the GPM interface for an FTP client process. The top pane shows a sequence of services: Start, FTP Client Begin Session Service, FTP Client CD Service, FTP Client PUT Service, FTP Client LIST Service, FTP Client End Session Service, and End. Callouts explain that the Begin and End services are used to open and close the session, and that the CD, PUT, and LIST services are used for the specific actions.

The bottom pane shows the configuration for the 'FTP Client Begin Session Service'. The configuration includes the following parameters:

Name	Value	Use JPATH?
CACertificateId	[No CA Certificates Available]	<input type="checkbox"/>
CipherStrength	STRONG	<input type="checkbox"/>
ClearControlChannel	No	<input type="checkbox"/>
ConnectionRetries	2	<input type="checkbox"/>
FTPClientAdapter	FTP Client Adapter	<input type="checkbox"/>
ProfileId	tradingpartner1	<input type="checkbox"/>
RemoteHost	000.00.000.00	<input type="checkbox"/>
RemotePasswd	*****	<input type="checkbox"/>
RemotePort		<input type="checkbox"/>

Callouts indicate that the configuration includes the FTP Client adapter and the specific services (CD, PUT, LIST) to be performed on the remote server.

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML:

```
<process name="default">
  <sequence>
```

```

<operation name="FTP Client Begin Session Service">
  <participant name="FTPClientBeginSession"/>
  <output message="FTPClientBeginSessionServiceTypeInputMessage">
    <assign to="CipherStrength">STRONG</assign>
    <assign to="ClearControlChannel">NO</assign>
    <assign to="ConnectionRetries">2</assign>
    <assign to="FTPClientAdapter">FTPClientAdapter</assign>
    <assign to="ProfileId">tradingpartner1</assign>
    <assign to="RemoteHost">000.00.000.00</assign>
    <assign to="RemotePasswd">uuuuuuuu</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="FTP Client CD Service">
  <participant name="FTPClientCd"/>
  <output message="FTPClientCdServiceTypeInputMessage">
    <assign to="CdUp">YES</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="FTP Client LIST Service">
  <participant name="FTPClientList"/>
  <output message="FTPClientListServiceTypeInputMessage">
    <assign to="ConnectionType">ACTIVE</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="FTP Client PUT Service">
  <participant name="FTPClientPut"/>
  <output message="FTPClientPutServiceTypeInputMessage">
    <assign to="DocumentId">doc12345</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="FTP Client End Session Service">
  <participant name="FTPClientEndSession"/>
  <output message="FTPClientEndSessionServiceTypeInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

```

```
</sequence>
</process>
```

Implementing the FTP Client adapter

To implement the FTP Client adapter, complete the following tasks:

1. Create a FTP Client adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the adapter. For information, see *Configuring the FTP Client adapter* on page 477.

Configuring the FTP Client adapter

To configure the FTP Client adapter, you must specify field settings in Application.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default)◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups, see <i>Managing Services and Adapters</i>.</p>
Perimeter Server	<p>Select the Perimeter server to use with this adapter. Default is node1 & local. Required.</p> <p>Note: You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring.</p>
Min Threads	A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be less than or equal to Max Threads value. Valid value is any integer. Default is 3. Required.
Max Threads	A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be greater than or equal to Min Threads value. Valid value is any integer. Default is 6. Required.
Local Control Port Range	Any valid port numbers not being used by another application running on the system. A port in the specified range will be used to establish a control channel to the remote FTP server. Optional.

Field	Description
Local Data Port Range	Any valid port numbers that are not being used by another application running on the system. Contact your network administrator if the port numbers are restricted in any way. A port in the specified range will be used to establish a data connection to the remote FTP server. Optional.

FTP Client Begin Session Service

The following table provides an overview of the FTP Client Begin Session service:

System name	FTP Client Begin Session Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	This service is used to start an FTP session with an external trading partner to exchange business documents.
Business usage	Use this service to establish a session with a trading partner FTP server.
Usage example	A business process is executed that translates a document that must be sent to a trading partner. After the translation, Application uses the FTP Client Begin Session service to establish a session with the trading partner's FTP server. The Begin Session service works through a configuration of the FTP Client adapter.
Preconfigured?	No
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client SITE service◆ FTP Client QUOTE service <p>To mask the values associated with the remote password parameter, use the Obscure Data - Process Data Values service in conjunction with the FTP Begin Session service. This service is presented in the GPM as Obscure Parameter on the All Services stencil.</p>
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a business process.

Business process context considerations	The FTP Client Begin Session service allows you to specify a remote password. To obscure this password in process data for the business process, you must use the Obscure Data - Process Data Values service within the same business process. The Obscure Data - Process Data Values service masks the values associated with parameters.
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	N/A
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = <code>admin</code> ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client Begin Session Service

To implement the FTP Client Begin Session service, complete the following tasks:

1. Create an FTP Client Begin Session service configuration. For information, see *Managing Services and Adapters*.
2. Use the FTP Client Begin Session service in a business process.

Configuring the FTP Client Begin Session Service

You can set the following values in the trading partner profile and specify it in the ProfileId field, or you can set these values in an instance of the service to only apply for that instance. If specified in the FTP Client Begin Session service, the following values override those in the FTP trading partner profile:

CACertificateId
 CipherStrength
 ConnectionRetries
 CharacterEncoding
 ConnectionTimeout
 RemoteHost
 RemotePasswd
 RemotePort
 RemoteUserId
 RetryDelay
 SSL
 SystemCertificateId

To configure the FTP Client Begin Session service, you must specify settings for the following fields in the GPM:

Field	Description
Name	Name this service will have in Application. Required.
Description	Description of service. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. (default) ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .
Config	Name of the service configuration. Select FTPClientBeginSession.

Field	Description
CACertificateId (trusted_root)	List of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID. Required if SSL = SSL_IMPLICIT or SSL_EXPLICIT. Obtain an SSL certificate from a Certificate Authority or from your trading partner. Check it into Application from the Admin menu selecting Trading Partner > Digital Certificates > CA to make it available in this list.
CipherStrength	The level of encryption to apply to the data that flows through the socket connection. Optional. Valid values are: <ul style="list-style-type: none"> ◆ ALL – WEAK or STRONG is accepted ◆ WEAK – 40 bit encryption is required ◆ STRONG – 128 bit or higher encryption is required (default)
ClearControlChannel	Indicates if information that travels across the control channel should be clear. Optional. Valid values are Yes and No.
ConnectionRetries	The number of times the service will try to connect to the trading partner system. Connection retries occur only with TCP/IP related issues. Optional. Valid value is any numeric value. While using the ConnectionRetries parameter, set the ResponseTimeout value to wait longer than the total time for RetryDelay and ConnectionRetires parameters. This setting allows the business process to remain active to perform the retries before the session times out and terminates. The following example illustrates the setting where the value of the ResponseTimeout (300) is greater than the total time taken by RetryDelay and ConnectionRetires parameters (30*5=150): <pre><assign to="ResponseTimeout">300</assign> <assign to="RetryDelay">30</assign> <assign to="ConnectionRetries">5</assign></pre>
CharacterEncoding	The encoding format used to encode all outgoing commands and incoming data. If CharacterEncoding is not specified, the default system encoding will be used. Valid value is any valid encoding scheme supported by Java. Optional.
ResponseTimeout	Maximum number of seconds the FTP Client Begin Session service waits for the trading partner system to respond before the session times out and terminates. You can also set this parameter from a trading profile. The value you specify in the FTP Client Begin Session service overrides the value you specify in the trading partner profile. Optional. Valid value is any numeric value. Default value is 30 seconds. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client Begin Session service resets the value to 1 second.
FTPClientAdapter	Select the FTP Client adapter for this service to use when beginning sessions with an FTP server. Required.
ProfileId	Trading partner profile identification. Optional. Valid value is any valid profile ID.
UsingRevealedPasswd	Indicates whether the password sent to the service is unobscured. Valid value is True or False. Default is False. Optional.
RemoteAccount	FTP remote login account. Valid value is valid login account. There is no default value. Optional.
RemoteHost	External trading partner host system (FTP server IP address or DNS name). Required. Valid value is a valid IP Address or DNS name.

Field	Description
RemotePasswd	FTP remote login password. Optional. Note: For the password to be masked in process data, the Obscure Data - Process Data Values service must also be used in the same business process. The name used to store the password must be the same as the specified RemoteUserId.
RemotePort	External trading partner port number. Required.
RemoteUserId	FTP remote login user name. Optional.
RetryDelay	The delay (in seconds) the adapter will wait before retrying. Optional. Valid value is any numeric value.
SSL	Determines SSL socket negotiation. Optional. Valid values are: <ul style="list-style-type: none"> ◆ SSL_IMPLICIT – FTP server expects and requires SSL to happen automatically at the time of connection. CACertificateId is required. ◆ SSL_EXPLICIT – FTP client requests SSL and a secure connection is negotiated. CACertificateId is required. ◆ SSL_NONE – Connection will not use SSL. (default)
SystemCertificateId	Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. This certificate confirms the identity of the client to the server. Required if SSL = SSL_IMPLICIT or SSL_EXPLICIT and the server requires client authentication. Obtain the certificate from your trading partner. Check it into Application from the Admin menu selecting Trading Partner > Digital Certificates > System to make it available in this list.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none"> ◆ erroronly – persists the transcript only when an error occurs ◆ on – always persists the transcript Default is on. Optional.

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client Begin Session service to the business process:

Parameter	Description
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server.
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client Begin Session service:

Parameter	Description
CACertificateId (trusted_root)	List of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID.
CipherStrength	The level of encryption to apply to the data that flows through the socket connection. Valid values are: <ul style="list-style-type: none">◆ ALL – WEAK or STRONG is accepted◆ WEAK – 40 bit encryption is required◆ STRONG – 128 bit or higher encryption is required
ClearControlChannel	Indicates if information that travels across the control channel should be clear. Valid values are Yes and No.
ConnectionRetries	The number of times the service will try to connect to the trading partner system. Valid value is any numeric value. Connection retries occur only with TCP/IP related issues. While using the ConnectionRetries parameter, set the ResponseTimeout value to wait longer than the total time for RetryDelay and ConnectionRetires parameters. This setting allows the business process to remain active to perform the retries before the session times out and terminates. The following example illustrates the setting where the value of the ResponseTimeout (300) is greater than the total time taken by RetryDelay and ConnectionRetires parameters (30*5=150): <assign to="ResponseTimeout">300</assign> <assign to="RetryDelay">30</assign> <assign to="ConnectionRetries">5</assign>
ResponseTimeout	Maximum number of seconds the FTP Client Begin Session service waits for the trading partner system to respond before the session times out and terminates. You can also set this parameter from a trading profile. The value you specify in the FTP Client Begin Session service overrides the value you specify in the trading partner profile. Optional. Valid value is any numeric value. Default value is 30 seconds. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client Begin Session service resets the value to 1 second.
FTPClientAdapter	Select the FTP Client adapter for this service to use when beginning sessions with an FTP server.
ProfileId	Trading partner profile identification. Valid value is any valid profile ID.
RemoteHost	External trading partner host system (FTP server IP address or DNS name). Valid value is a valid IP Address or DNS name.
RemotePasswd	FTP remote login password.
RemotePort	External trading partner port number.
RemoteUserId	FTP remote login user name.
RetryDelay	The delay (in seconds) the adapter will wait before retrying. Valid value is any numeric value.

Parameter	Description
SSL	The SSL flag that determines SSL socket negotiation. Valid values are: <ul style="list-style-type: none"> ◆ SSL_IMPLICIT – FTP server expects and requires SSL to happen automatically at the time of connection. ◆ SSL_EXPLICIT – FTP client requests SSL and a secure connection is negotiated. ◆ SSL_NONE – Connection will not use SSL.
SystemCertificateId	Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. Valid value is any alphanumeric string.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none"> ◆ erroronly – persists the transcript only when an error occurs ◆ on – always persists the transcript

Business Process Example

The following example business process illustrates using the FTP Client Begin Session service:

```
<process name="FtpExample">
  <sequence>

    <operation name="Obscure Password">
      <!-- insert obscured password into process data -->
      <participant name="FTPClientObscureParameter"/>
      <output message="outmsg">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="FTP Client Begin Session Service">
      <participant name="FTPClientBeginSession"/>
      <output message="FTPClientBeginSessionServiceTypeInputMessage">
        <assign to="FTPClientAdapter">FTPClientAdapter</assign>
        <assign to="RemoteHost">hostb</assign>
        <assign to="RemoteUserId">admin</assign>
        <!-- copy obscured password from process data to service -->
        <assign to="RemotePasswd" from="admin/text()"></assign>
        <assign to="RemotePort">30651</assign>

        <assign to="CipherStrength">STRONG</assign>
        <assign to="SSL">SSL_MUST</assign>
        <assign to="CACertificateId">FTP Server CA Cert</assign>
        <assign to="SystemCertificateId">FtpClientSystemCert</assign>

        <assign to="RemoteUserId">admin</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">

```

```

        <assign to="FTPClientBeginSessionServiceResults" from="*"></assign>
    </input>
</operation>
[[end session here]]
</process>

```

The following example shows how to use the revealObscured function when the user ID contains domain or special characters. First, create a name in the Obscure Service without any special characters and assign the appropriate password to it. In the following example, the name created in the Obscure Service is "abcd" while the userid is sgp-abcd\abcd.

The parameter to the revealObscured() function is the node containing the obscured password. The function uses the node name as the key and the node value as the obscured password when unobscuring.

```

<operation name="Obscure FTP client password">
    <participant name="FTPClientObscureParameter"/>
    <output message="outmsg">
        <assign to="." from="*" />
    </output>
    <input message="inmsg">
        <assign to="ObscureResult" from="*" />
    </input>
</operation>

<operation name="PS FTP BEGIN SESSION SERVICE">
    <participant name="FTPClientBeginSession"/>
    <output message="BeginSessionRequest">
        .....
        <assign to="RemoteUserId">sgp-abcd\abcd</assign>
        <assign to="UsingRevealedPasswd">true</assign>
        <assign to="RemotePasswd" from="revealObscured(ObscureResult/abcd)"/>
        .....
    </output>
    <input message="inmsg">
        <assign to="FtpBeginSessionServiceResults" from="*" />
    </input>
</operation>

```

FTP Client CD Service

The following table provides an overview of the FTP Client CD service:

System name	FTP Client CD Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	This service is used to change directories on the trading partner's FTP server.
Business usage	Use this service to change directories on the trading partner FTP server system.
Usage example	A Application business process is executed that requires retrieving a document from the trading partner's system. The document is located in a directory other than the home directory. Application uses the FTP Client CD service, working through the FTP Client adapter, to change directories to the one where the document is located. Application can then use the FTP Client GET service to retrieve the document.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client SITE service◆ FTP Client QUOTE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is initiated from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 0–Success ◆ 1–Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client CD Service

To implement the FTP Client CD service, complete the following tasks:

1. Create an FTP Client CD service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client CD service. For information, see *Configuring the FTP Client CD Service* on page 489.

3. Use the FTP Client CD service in a business process.

Configuring the FTP Client CD Service

To configure the FTP Client CD service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
CdUp	<ul style="list-style-type: none">◆ YES-Causes the FTP protocol command CDUP to be sent to the server, which results in the current directory moving one level up.◆ NO-Causes the FTP protocol command CWD + Directory to be sent to the server. Default is NO.
Directory	The directory to change to. Can also be a relative directory (like ..) depending on server capabilities. Required if CdUp is No. Note: If CdUp=YES, this parameter is ignored.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client CD service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client CD service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client CD service:

Parameter	Description
CdUp	<ul style="list-style-type: none">◆ YES-Causes the FTP protocol command CDUP to be sent to the server, which results in the current directory moving one level up.◆ NO-Causes the FTP protocol command CWD + Directory to be sent to the server.
Directory	The directory to change to. Can also be a relative directory (like ..) depending on server capabilities. Required if CdUp is NO. Note: If CdUp=YES, Directory will be ignored.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client CD service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Business Process Example

The following example business process illustrates using the FTP Client CD service:

```
<sequence>
```

```
  [[Insert FTP Client Begin Session]]
```

```
  <operation name="FTP CD SERVICE">
    <participant name="FTPClientCd"/>
    <output message="CdRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="Directory">home/username/documents</assign>
    </output>
    <input message="inmsg">
      <assign to="FTPClientCdResults" from="*"></assign>
    </input>
  </operation>
```

```
  [[ Insert FTP Client End Session ]]
```

</sequence>

FTP Client DELETE Service

The following table provides an overview of the FTP Client DELETE service:

System name	FTP Client DELETE Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	Use to delete a document in a specified directory on the trading partner's FTP server.
Business usage	Use this service to delete a document on the trading partner system when FTP is used.
Usage example	A Application business process is executed that requires the removal of a document on the trading partner's system. Application uses the FTP Client DELETE service, working through the FTP Client adapter, to remove the specified document from the directory on the trading partner system.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client SITE service◆ FTP Client QUOTE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 0–Success ◆ 1–Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client DELETE Service

To implement the FTP Client DELETE service, complete the following tasks:

1. Create an FTP Client DELETE service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client DELETE service. For information, see *Configuring the FTP Client DELETE Service* on page 494.

3. Use the FTP Client DELETE service in a business process.

Configuring the FTP Client DELETE Service

To configure the FTP Client DELETE service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
RemoteFileName	The name of the file to delete from the remote trading partner directory. Optional.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client DELETE service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client DELETE service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client DELETE service:

Parameter	Description
RemoteFileName	The name of the file to delete from the remote trading partner directory. Optional.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client DELETE service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Business Process Example

The following example business process illustrates using the FTP Client DELETE service.

```
<sequence>

  [[ Insert FTP Client Begin Session ]]

  <operation name="FTP DELETE SERVICE">
    <participant name="FTPClientDelete"/>
    <output message="DeleteRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="RemoteFileName">Filename.txt</assign>
    </output>
    <input message="inmsg">
      <assign to="FtpGetServiceResults" from="*"></assign>
    </input>
  </operation>

  [[ Insert FTP Client End Session ]]

</sequence>
```

FTP Client End Session Service

The following table provides an overview of the FTP Client End Session service:

System name	FTP Client End Session Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	This service is used to end an FTP session with an external trading partner FTP server.
Business usage	You would use this service as the last functional activity in a business process that sends an FTP request to a trading partner. This service can only be used if the FTP Client Begin Session service was used previously in the business process.
Usage example	A Application business process is executed that translates a document that must be sent to a trading partner. After the translation, Application begins a session with the trading partner using the FTP Client adapter, sends the document, then ends the session using the FTP Client End Session service.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client SITE service◆ FTP Client QUOTE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service can be invoked in a business process to terminate a session with an FTP server.

Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ 0–Success ◆ 1–Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client End Session Service

To implement the FTP Client End Session service, complete the following tasks:

1. Create an FTP Client End Session service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.

2. Configure the FTP Client End Session service. For information, see *Configuring the FTP Client End Session Service* on page 498.
3. Use the FTP Client End Session service in a business process.

Configuring the FTP Client End Session Service

To configure the FTP Client End Session service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
ResponseTimeout	<p>Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates.</p> <p>Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client End Session service resets the value to 1 second.</p>
SessionToken	<p>Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required.</p> <p>Note: The session token is returned from the FTP Client Begin Session service.</p>
SaveTranscript	<p>Indicates how to handle the transcript. Valid values are:</p> <ul style="list-style-type: none"> ◆ erroronly – persists the transcript only when an error occurs ◆ on – always persists the transcript <p>Default is on. Optional.</p>

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client End Session service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client End Session service:

Parameter	Description
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client End Session service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Business Process Example

The following example business process illustrates using the FTP Client End Session service:

```
<sequence>
  [[Insert FTP Client Begin Session]]
  <operation name="FTP END SESSION SERVICE">
    <participant name="FTPClientEndSession"/>
    <output message="EndSessionRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
    </output>
    <input message="inmsg">
      <assign to="FtpEndSessionResults" from="*"></assign>
    </input>
  </operation>
</sequence>
```

FTP Client GET Service

The following table provides an overview of the FTP Client GET service:

System name	FTP Client GET Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	This service is used to retrieve one or more documents from a specified directory on the trading partner's FTP server.
Business usage	You would use this service to retrieve one or more documents from a trading partner and move them into Application when the FTP protocol is required as the transport mechanism.
Usage example	A Application business process is executed that must retrieve a specified file from the external trading partner. Application uses the FTP Client GET service, working through the FTP Client adapter, to retrieve the file from a specified directory on the trading partner system.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client QUOTE service◆ FTP Client SITE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is initiated from a business process.

Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service – If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service – This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client GET Service

To implement the FTP Client GET service, complete the following tasks:

1. Create an FTP Client GET service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.

2. Configure the FTP Client GET service. For information, see *Configuring the FTP Client GET Service* on page 502.
3. Use the FTP Client GET service in a business process.

Configuring the FTP Client GET Service

To configure the FTP Client GET service, you must specify settings for the following fields in the UI or the GPM:

Field	Description
Name	Name this adapter will have in Application
Description	Description of adapter
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Config	Name of the service configuration.
CheckFileSize	<p>Use to check that the file size is stable before downloading the file. Valid values are YES and NO. NO is the default value. When YES is specified, the FTP GET service checks the file size at 5-second intervals. The FTP GET service downloads the file only when the file size remains unchanged. Any change in the file size indicates that the file is still being transferred to the server and the service sends an error to the business process. Optional. You cannot use this parameter if RemoteFilePattern is specified.</p> <p>Note: This parameter is supported with the following FTP servers:</p> <ul style="list-style-type: none"> ◆ Windows IIS FTP server ◆ Standard UNIX FTP server – such as SUN Solaris, HP-UP, AIX ◆ Standard LINUX FTP server – Redhat ◆ War FTP Daemon 1.70/80 series (Windows)
ConnectionType	<p>Value that describes which way the data connection will be made when the data is transferred. Optional. Value values are:</p> <ul style="list-style-type: none"> ◆ ACTIVE — The server will make the connection. (Default) ◆ PASSIVE — The adapter will make the connection.
ListNamesErrorSetSuccess	Ignore 550 error code when executing the NLST command. Optional. Valid values are YES and NO.

Field	Description
RemoteFileName	<p>The name of the file to be retrieved from the remote trading partner. Optional. You cannot use this parameter if RemoteFilePattern is specified.</p> <p>Note: Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank.</p>
RemoteFilePattern	<p>The file filter pattern. Using this field activates multiple-get mode. Optional. You cannot use this parameter if RemoteFileName is specified.</p> <p>Note: Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank.</p>
RepresentationType	<p>The FTP representation type that will be used for the transfer. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ ASCII – transfers the data in ASCII mode ◆ BINARY – transfers the data in binary mode (default)
ResponseTimeout	<p>Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates.</p> <p>Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client GET service resets the value to 1 second.</p>
RetrieveErrorSetSuccess	<p>Ignore 550 error code when executing the RETR command. Optional. Valid values are YES and NO.</p>
SaveTranscript	<p>Indicates how to handle the transcript. Valid values are:</p> <ul style="list-style-type: none"> ◆ erroronly – persists the transcript only when an error occurs ◆ on – always persists the transcript <p>Default is on. Optional.</p>
DelayWaitingOnIO	<p>Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service is operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional.</p>
SessionToken	<p>Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.</p> <p>Note: The session token is returned from the FTP Client Begin Session service.</p>

Parameters Passed from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client GET service:

Parameter	Description
CheckFileSize	<p>Use to check that the file size is stable before downloading the file. Valid values are YES and NO. NO is the default value. When YES is specified, the FTP GET service checks the file size at 5-second intervals. The FTP GET service downloads the file only when the file size remains unchanged. Any change in the file size indicates that the file is still being transferred to the server and the service sends an error to the business process. Optional. You cannot use this parameter if RemoteFilePattern is specified.</p> <p>Note: This parameter is supported with the following FTP servers:</p> <ul style="list-style-type: none">◆ Windows IIS FTP server◆ Standard UNIX FTP server – such as SUN Solaris, HP-UP, AIX◆ Standard LINUX FTP server – Redhat◆ War FTP Daemon 1.70/80 series (Windows)
ConnectionType	<p>Value that describes which way the data connection will be made when the data is transferred. Optional. Value values are:</p> <ul style="list-style-type: none">◆ ACTIVE — The server will make the connection. (Default)◆ PASSIVE — The adapter will make the connection.
ListNamesErrorSetSuccess	<p>Ignore 550 error code when executing the NLST command. Optional. Valid values are YES and NO.</p>
RemoteFileName	<p>The name of the file to be retrieved from the remote trading partner. If a value is entered in this field, RemoteFilePattern cannot be used. Optional. You cannot use this parameter if RemoteFilePattern is specified.</p> <p>Note: Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank.</p>
RemoteFilePattern	<p>The file filter pattern. Using this field activates multiple-get mode. If a value is entered in this field, then RemoteFileName cannot be used. Optional. You cannot use this parameter if RemoteFileName is specified.</p> <p>Note: Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank.</p>
RepresentationType	<p>The FTP representation type that will be used for the transfer. Optional. Valid values are:</p> <ul style="list-style-type: none">◆ ASCII – transfers the data in ASCII mode◆ BINARY – transfers the data in binary mode (default)

Parameter	Description
ResponseTimeout	<p>Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates.</p> <p>Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client GET service resets the value to 1 second.</p>
RetrieveErrorSetSuccess	Ignore 550 error code when executing the RETR command. Optional. Valid values are YES and NO.
SaveTranscript	<p>Indicates how to handle the transcript. Valid values are:</p> <ul style="list-style-type: none"> ◆ erroronly – persists the transcript only when an error occurs ◆ on – always persists the transcript <p>Default is on. Optional.</p>
DelayWaitingOnIO	Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service is operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional.
SessionToken	<p>Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.</p> <p>Note: The session token is returned from the FTP Client Begin Session service.</p>

Parameters Passed from Service to Business Process

The following table contains the parameters passed from the FTP Client GET service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required.
DocumentList	<p>Provides a list of document IDs that were created for the files retrieved by the FTP Client GET service. Required.</p> <p>Note: If a single document was retrieved, the service will place it as the primary document.</p>

Business Process Examples

The following example business processes illustrate using commands supported by the FTP Client GET service.

This process gets a binary file named TestDoc using the passive connection type from the server:

```
<sequence>

[[Insert FTP Client Begin Session here]]

<operation name="FTP GET SERVICE">
  <participant name="FTPClientGet"/>
  <output message="GetRequest">
    <assign to="SessionToken"
      from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
    </assign>
    <assign to="RemoteFileName">TestDoc</assign>
    <assign to="ConnectionType">PASSIVE</assign>
    <assign to="RepresentationType">BINARY</assign>
  </output>
  <input message="inmsg">
    <assign to="FtpGetServiceResults" from="*"></assign>
  </input>
</operation>

[[Insert FTP Client End Session here]]

</sequence>
```

This process illustrates using a multiple GET command:

```
<sequence>

[[Insert FTP Client Begin Session here]]

<operation name="FTP MULTIPLE GET SERVICE">
  <participant name="FTPClientGet"/>
  <output message="GetRequest">
    <assign to="SessionToken"
      from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
    </assign>
    <assign to="RemoteFilePattern">*.txt</assign>
    <assign to="ConnectionType">PASSIVE</assign>
    <assign to="RepresentationType">BINARY</assign>
  </output>
  <input message="inmsg">
    <assign to="FtpGetServiceResults" from="*"></assign>
  </input>
</operation>

[[Insert FTP Client End Session here]]

</sequence>
```

The following example business process illustrates using an implicit assign to add a message from the FTP Client GET service to the process data:

```
<input message="inmsg">
  <assign to="." from="*"></assign>
</input>
```

The following example business process illustrates using an explicit assign to add a message from the FTP Client GET service to the process data:

```
<input message="inmsg">
  <assign to="StatusReport" from="Status_Rpt (&apos;StatusReport&apos;)"></assign>
  <assign to="FTPGetResults" from="*">
</input>
```

Application supports either implicit assign or explicit assign, but not both at the same time, for example:

```
<input message="inmsg">
  <assign to="StatusReport" from="Status_Rpt (&apos;StatusReport&apos;)"></assign>
  <assign to="." from="*"></assign>
</input>
```

FTP Client LIST Service

The following table provides an overview of the FTP Client LIST service:

System name	FTP Client LIST Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	Use this service to retrieve a list of documents in a specified directory on the trading partner's FTP server.
Business usage	You would use this service to retrieve a list of files in a specified directory on the trading partner's system and return the list to Application when the FTP protocol is required as the transport mechanism.
Usage example	A Application business process is executed that must retrieve a list of files from the external trading partner. Application uses the FTP Client LIST service, working through the FTP Client adapter, to retrieve the list of files from a specified directory on the trading partner system.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client SITE service◆ FTP Client QUOTE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a business process.

Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner.</p>

Implementing the FTP Client LIST Service

To implement the FTP Client LIST service:

1. Create an FTP Client LIST service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client LIST service. For information, see *Configuring the FTP Client LIST Service* on page 510.
3. Use the FTP Client LIST service in a business process.

Configuring the FTP Client LIST Service

To configure the FTP Client LIST service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
ConnectionType	Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are: <ul style="list-style-type: none">◆ ACTIVE - The server will make the connection. (Default)◆ PASSIVE - The adapter will make the connection.
NamesOnly	Whether to include just names or all information to the server. Optional. <ul style="list-style-type: none">◆ YES – Causes the FTP protocol command NLST to be sent to the server. The results will be a stream of names of files and no other information.◆ NO – Causes the FTP protocol command LIST to be sent to the server. The results will vary in form and information depending on the server. (Default)
RemoteFileName	File name or pattern to do the listing for. If not included the listing will be for a *.* pattern (depending on server implementation). Optional.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client LIST service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Optional. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript (Default)

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client LIST service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required.

Parameter	Description
ListNames	Provides information about the files included in the specified directory. Optional. Only returned if NamesOnly=YES.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client LIST service:

Parameter	Description
ConnectionType	Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are: <ul style="list-style-type: none"> ◆ ACTIVE - The server will make the connection. (Default) ◆ PASSIVE - The adapter will make the connection.
NamesOnly	Whether to include just names or all information to the server. Optional. <ul style="list-style-type: none"> ◆ YES – Causes the FTP protocol command NLST to be sent to the server. The results will be a stream of names of files and no other information. ◆ NO – Causes the FTP protocol command LIST to be sent to the server. The results will vary in form and information depending on the server. (Default)
ListNamesErrorSetSuccess	Ignore 550 error code when executing the NLST command. Optional. Valid values are YES and NO.
RemoteFileName	File name or pattern to do the listing for. If not included the listing will be for a *.* pattern (depending on server implementation). Optional.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client LIST service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Optional. Valid values are: <ul style="list-style-type: none"> ◆ erroronly – persists the transcript only when an error occurs ◆ on – always persists the transcript (Default)

Business Process Example

The following example business processes illustrate the use of the FTP Client LIST service:

Example of a Names Only listing:

```

<sequence>

  [[ Insert FTP Client Begin Session ]]

  <operation name="FTP LIST SERVICE NAME">
    <participant name="FTPClientList"/>
    <output message="ListRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="NamesOnly">YES</assign>
      <assign to="RemoteFileName">*.txt</assign>
      <assign to="ConnectionType">PASSIVE</assign>
    </output>
    <input message="inmsg">
      <assign to="NameListResults" from="*"></assign>
    </input>
  </operation>

  [[ Insert FTP Client End Session ]]

</sequence>

```

Example of a Raw listing:

```

<sequence>

  [[ Insert FTP Client Begin Session ]]

  <operation name="FTP LIST SERVICE RAW">
    <participant name="FTPClientList"/>
    <output message="ListRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="NamesOnly">NO</assign>
      <assign to="ConnectionType">PASSIVE</assign>
    </output>
    <input message="inmsg">
      <assign to="RawListResults" from="*"></assign>
    </input>
  </operation>

  [[ Insert FTP Client End Session ]]

</sequence>

```

FTP Client MOVE Service

The following table provides an overview of the FTP Client MOVE service:

System name	FTP Client MOVE Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	This service is used to rename a document or move it from one directory to another.
Business usage	You would use this service to rename or move a document on a trading partner's system when the FTP protocol is required as the communication mechanism with the trading partner.
Usage example	A Application business process is executed that requires moving a document on the trading partner's system. Application uses the FTP Client MOVE service, working through the FTP Client adapter, to move the specified document from one directory to another on the trading partner system.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client SITE service◆ FTP Client QUOTE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 0–Success ◆ 1–Error
Restrictions	No
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client MOVE Service

To implement the FTP Client MOVE service, complete the following tasks:

1. Create an FTP Client MOVE service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client MOVE service. For information, see *Configuring the FTP Client MOVE Service* on page 515.

3. Use the FTP Client MOVE service in a business process.

Configuring the FTP Client MOVE Service

To configure the FTP Client MOVE service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
RemoteFromFileName	The current name of the remote file. Required.
RemoteToFileName	The new name of the remote file. Required.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client MOVE service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client MOVE service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client MOVE service:

Parameter	Description
RemoteFromFileName	The current name of the remote file. Required.
RemoteToFileName	The new name of the remote file. Required.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client MOVE service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Business Process Example

The following example business process illustrates using the FTP Client MOVE service:

```
<sequence>

  [[Insert FTP Client Begin Session ]]

  <operation name="FTP MOVE SERVICE">
    <participant name="FTPClientMove"/>
    <output message="MoveRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="RemoteFromFileName">oldFileNameAndDirectory</assign>
      <assign to="RemoteToFileName">newFileNameAndDirectory</assign>
    </output>
    <input message="inmsg">
      <assign to="FtpMoveResults" from="*"></assign>
    </input>
  </operation>

  [[Insert FTP Client End Session]]

</sequence>
```

FTP Client PUT Service

The following table provides an overview of the FTP Client PUT service:

System name	FTP Client PUT Service
Graphical Process Modeler (GPM) categories	All Services
Description	Used to place a document or documents in a specified directory on the trading partner's FTP server.
Business usage	Use this service to transfer a document or documents from Application to a trading partner when the FTP protocol is required as the transport.
Usage example	You run a business process that translates a document that must be sent to a trading partner. After the translation, Application uses the FTP Client PUT service, working through the FTP Client adapter, to place the document in a specified directory on the trading partner system.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client GET service◆ FTP Client LIST service◆ FTP Client MOVE service◆ FTP Client PWD service◆ FTP Client QUOTE service◆ FTP Client SITE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration. 2. Find FTP Server Adapter. 3. If not already selected, select the Enabled check box. <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager. 2. Find the FTPClientDemoAllServices business process. 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code> 4. Verify that the business process runs successfully. <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP PUT Client Service

To implement the FTP Client PUT service, complete the following tasks:

1. Create an FTP Client PUT service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client PUT service. For information, see *Configuring the FTP Client PUT Service* on page 520.

3. Use the FTP Client PUT service in a business process.

Configuring the FTP Client PUT Service

To configure the FTP Client PUT service, you must specify settings for the following fields in the UI or the GPM:

Field	Description
Name	Name this adapter will have in Application
Description	Description of adapter
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .
Config	Name of the service configuration.
ConnectionType	Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are: <ul style="list-style-type: none">◆ ACTIVE - The server will make the connection. (Default)◆ PASSIVE - The adapter will make the connection.
DocumentId	Document ID to PUT to the remote server. A single DocumentId can appear directly in the message to the service or any number of DocumentIds can appear under the DocumentList element. Optional. Note: The FTP Client PUT service will use DocumentList if a list is provided. If no list is specified in DocumentList, the service will use DocumentId. The service will not use both DocumentList and DocumentId. If no values are specified for either DocumentList or DocumentId, the service will PUT the primary document to the remote server.
DocumentList	List of documents to PUT to the remote server. Each item must be a DocumentId. A list could look like the following example: <DocumentList> <DocumentId>12345</DocumentId> <DocumentId>67890</DocumentId> </DocumentList>
RemoteFileName	Filename that Application uses to place the document on the remote system. If not specified, the name of the document will be used. Do not use this parameter if putting multiple documents on a server. Optional.
RepresentationType	Representation used for transferring the file. Optional. Valid values: <ul style="list-style-type: none">◆ ASCII – Transfers the data in ascii ASCII mode◆ BINARY – Transfers the data in binary mode (default)

Field	Description
ResponseTimeout	<p>Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates.</p> <p>Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client PUT service resets the value to 1 second.</p>
DelayWaitingOnIO	<p>Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional.</p>
SaveTranscript	<p>Indicates how to handle the transcript. Valid values are:</p> <ul style="list-style-type: none"> ◆ erroronly – Persists the transcript only when an error occurs ◆ on – Always persists the transcript <p>Default is on. Optional.</p>
SessionToken	<p>Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.</p> <p>Note: The session token is returned from the FTP Client Begin Session service.</p>
AppendMode	<p>Specifies whether to perform an append operation. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ YES - Append to the destination file ◆ NO - Create a new file <p>To enter this parameter, select Advanced in the GPM Service configuration pane. Select New, and type</p> <p>Name = AppendMode Value = YES</p>
UseDocBodyName	<p>Specifies whether to use document body name as the remote file name. This parameter is only use in MPUT operation. Valid values are:</p> <ul style="list-style-type: none"> ◆ YES – Use document body name ◆ NO – Use document name <p>Default is NO. Optional.</p>

Parameters Passed from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client PUT service:

Parameter	Description
ConnectionType	<p>Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ ACTIVE -The server will make the connection. (Default) ◆ PASSIVE - The adapter will make the connection.

Parameter	Description
DocumentId	<p>Document ID to PUT to the remote server. A single DocumentId can appear directly in the message to the service or any number of DocumentIds can appear under the DocumentList element. Optional.</p> <p>Note: The FTP Client PUT service will use DocumentList if a list is provided. If no list is specified in DocumentList, the service will use DocumentId. The service will not use both DocumentList and DocumentId. If no values are specified for either DocumentList or DocumentId, the service will PUT the primary document to the remote server.</p>
DocumentList	<p>List of documents to PUT to the remote server. Each item must be a DocumentId. A list could look like the following example:</p> <pre><DocumentList> <DocumentId>12345</DocumentId> <DocumentId>67890</DocumentId> </DocumentList></pre>
RemoteFileName	<p>Filename that Application uses to place the document on the remote system. If not specified, the name of the document will be used. Do not use this parameter if putting multiple documents on a server. Optional.</p>
RepresentationType	<p>Representation used for transferring the file. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ ASCII – Transfers the data in ascii ASCII mode ◆ BINARY – Transfers the data in binary mode (default)
ResponseTimeout	<p>Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates.</p> <p>Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client PUT service resets the value to 1 second.</p>
DelayWaitingOnIO	<p>Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service is operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional.</p>
SaveTranscript	<p>Indicates how to handle the transcript. Valid values are:</p> <ul style="list-style-type: none"> ◆ erroronly – Persists the transcript only when an error occurs ◆ on – Always persists the transcript <p>Default is on. Optional.</p>
SessionToken	<p>Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.</p> <p>Note: The session token is returned from the FTP Client Begin Session service.</p>

Parameter	Description
AppendMode	<p>Specifies whether to perform an append operation. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ YES - Append to the destination file ◆ NO - Create a new file <p>To enter this parameter, select Advanced in the GPM Service configuration pane. Select New, and type Name = AppendMode Value = YES</p>
UseDocBodyName	<p>Specifies whether to use document body name as the remote file name. This parameter is only use in MPUT operation. Valid values are:</p> <ul style="list-style-type: none"> ◆ YES – Use document body name ◆ NO – Use document name <p>Default is NO. Optional.</p>

Parameters Passed from Service to Business Process

The following table contains the parameters passed from the FTP Client PUT service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server.

Business Process Examples

The following business process uses the FTP Client adapter to send all documents received from a GET operation from Application to the remote FTP server:

```
<sequence>

  [[Insert FTP Client Begin Session]]

  <operation name="Get">
    <participant name="FTPClientGet"/>
    <output message="GetRequest">
      <assign to="SessionToken"
        from="/ProcessData/BeginSessionResults/SessionToken/text()">
      </assign>
      <assign to="RemoteFilePattern">*. *</assign>
    </output>
    <input message="GetResults">
      <assign to="GetResults" from="DocumentList"/>
    </input>
  </operation>
```

```

<operation name="Put">
  <participant name="FTPClientPut"/>
  <output message="PutRequest">
    <assign to="SessionToken"
      from="/ProcessData/BeginSessionResults/SessionToken/text()">
    </assign>
    <assign to="." From="/ProcessData/GetResults/DocumentList"/>
    <assign to="RepresentationType">ASCII</assign>
  </output>
  <input message="FtpPutResults">
    <assign to="PutResults" from="*"></assign>
  </input>
</operation>

```

```
[[Insert FTP Client End Session]]
```

```
</sequence>
```

The following example uses the FTP Client adapter to send all documents in a Document list from Application to the remote FTP server:

```
<sequence>
```

```
[[Insert FTP Client Begin Session]]
```

```

<operation name="Put">
  <participant name="FTPClientPut"/>
  <output message="PutRequest">
    <assign to="SessionToken"
      from="/ProcessData/BeginSessionResults/SessionToken/text()">
    </assign>
    <assign to="DocumentList/DocumentId" append="yes">12345</assign>
    <assign to="DocumentList/DocumentId" append="yes">67890</assign>
    <assign to="RepresentationType">ASCII</assign>
  </output>
  <input message="FtpPutResults">
    <assign to="PutResults" from="*"></assign>
  </input>
</operation>

```

```
[[Insert FTP Client End Session]]
```

```
</sequence>
```

The following example uses the append function:

```

<operation name="FTP Client PUT Service">
  <participant name="FTPClientPut"/>
  <output message="FTPClientPutServiceTypeInputMessage">
    <assign to="RemoteFileName">appenddoc</assign>
    <assign to="AppendMode">YES</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>

```

```
</operation
```

The following example uses the FTP Client adapter to send the primary document from Application to the remote FTP server:

```
<sequence>
```

```
[[Insert FTP Client Begin Session]]
```

```
<operation name="FTP PUT SERVICE">  
  <participant name="FTPClientPut"/>  
  <output message="PutRequest">  
    <assign to="SessionToken"  
      from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">  
    </assign>  
    <assign to="RemoteFileName">TestDoc</assign>  
    <assign to="ConnectionType">PASSIVE</assign>  
    <assign to="RepresentationType">BINARY</assign>  
    <assign to="." From="PrimaryDocument"></assign>  
  </output>  
  <input message="inmsg">  
    <assign to="FtpPutServiceResults" from="*"></assign>  
  </input>  
</operation>
```

```
[[Insert FTP Client End Session]]
```

```
</sequence>
```

FTP Client PWD Service

The following table provides an overview of the FTP Client PWD service:

System name	FTP Client PWD Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	This service is used to print the working directory on the trading partner's FTP server.
Business usage	You would use this service to get information about the current working directory on the trading partner FTP server.
Usage example	A Application business process is executed that places a document in a specific directory on the trading partner's system. The business process writer wants to ensure that status information within the business process includes the name of the directory where the document was placed. The business process writer includes the FTP Client PWD service in the process definition and the service places the directory information in the business process.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client SITE service◆ FTP Client QUOTE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a business process.

Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ 0–Success ◆ 1–Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client PWD Service

To implement the FTP Client PWD service, complete the following tasks:

1. Create an FTP Client PWD service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.

2. Configure the FTP Client PWD service. For information, see *Configuring the FTP Client PWD Service* on page 528.
3. Use the FTP Client PWD service in a business process.

Configuring the FTP Client PWD Service

To configure the FTP Client PWD service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
ResponseTimeout	<p>Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates.</p> <p>Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client PWD service resets the value to 1 second.</p>
SessionToken	<p>Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.</p> <p>Note: The session token is returned from the FTP Client Begin Session service.</p>
SaveTranscript	<p>Indicates how to handle the transcript. Valid values are:</p> <ul style="list-style-type: none"> ◆ erroronly – persists the transcript only when an error occurs ◆ on – always persists the transcript <p>Default is on. Optional.</p>

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client PWD service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. The text includes the server-specific text indicating the current working directory for the session. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client PWD service:

Parameter	Description
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client PWD service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Business Process Example

The following example business process illustrates using the FTP Client PWD service:

```
<sequence>

  [[ Insert FTP Client Begin Session ]]

  <operation name="FTP PWD SERVICE">
    <participant name="FTPClientPwd"/>
    <output message="PwdRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
    </output>
    <input message="inmsg">
      <assign to="FTPClientPwdResults" from="*"></assign>
    </input>
  </operation>

  [[ Insert FTP Client End Session ]]

</sequence>
```

FTP Client QUOTE Service

The following table provides an overview of the FTP Client QUOTE service:

System name	FTP Client QUOTE Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	The FTP Client QUOTE service is used to pass custom commands to the trading partner's FTP server.
Business usage	You would use this service to pass commands that the FTP server can support, but which are not included in the standard protocol.
Usage example	A Application business process is executed that translates a document that must be sent to a trading partner. After the translation, Application establishes a session with the trading partner FTP server (using the FTP Client adapter), and uses the QUOTE service to issue a print command on the host.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client SITE service
Application requirements	An FTP server at the external trading partner location.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client QUOTE Service

To implement the FTP Client QUOTE service, complete the following tasks:

1. Create an FTP Client QUOTE service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client QUOTE service. For information, see *Configuring the FTP Client QUOTE Service* on page 532.

3. Use the FTP Client QUOTE service in a business process.

Configuring the FTP Client QUOTE Service

To configure the FTP Client QUOTE service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
QuoteCommand	Enter the exact text of the command that will be run on the server (do not include the word 'Quote'). Required. Valid value is any text to send to the server.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client QUOTE service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client QUOTE service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. If the exchange involves a waiting state, the final transcript document ID contains the record of the full exchange.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client QUOTE service:

Parameter	Description
QuoteCommand	Enter the exact text of the command that will be run on the server (do not include the word 'Quote'). Required. Valid value is any text to send to the server.
ResponseTimeout	Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client QUOTE service resets the value to 1 second.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Business Process Example

The following example business process illustrates sending a site command using the FTP Client QUOTE service:

```
<sequence>

  [[ Insert FTP Client Begin Session ]]

  <operation name="FTP QUOTE SERVICE">
    <participant name="FTPClientQuote"/>
    <output message="QuoteRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="QuoteCommand">SITE HELP</assign>
    </output>
    <input message="inmsg">
      <assign to="FTPClientQuoteResults" from="*"></assign>
    </input>
  </operation>

  [[ Insert FTP Client Begin Session ]]

</sequence>
```

FTP Client SITE Service

The following table provides an overview of the FTP Client SITE service:

System name	FTP Client SITE Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > FTP Client
Description	This service is used to send site-specific control commands to an FTP server.
Business usage	You would use this service to send site-specific control commands to a trading partner FTP server.
Usage example	A Application business process is executed that must send a site-specific control command to a trading partner. Application uses the FTP Client SITE service to send a site-specific control command to the trading partner's FTP server.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ FTP Client adapter◆ FTP Client Begin Session service◆ FTP Client CD service◆ FTP Client DELETE service◆ FTP Client End Session service◆ FTP Client LIST service◆ FTP Client GET service◆ FTP Client MOVE service◆ FTP Client PUT service◆ FTP Client PWD service◆ FTP Client QUOTE service
Application requirements	An FTP server at the external trading partner location. Information about this server must be configured in the trading profile.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 0–Success ◆ 1–Error
Restrictions	None
Persistence level	Default
Testing considerations	<p>Test this service by running the FTPClientDemoAllServices business process provided with Application. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.</p> <p>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Deployment > Services > Configuration 2. Find FTP Server Adapter 3. If not already selected, select the Enabled check box <p>To test this service, perform the following steps from the Application Admin Console:</p> <ol style="list-style-type: none"> 1. Choose Business Processes > Manager 2. Find the FTPClientDemoAllServices business process 3. Run the FTPClientDemoAllServices business process with the following settings: <ul style="list-style-type: none"> ◆ Run As User = Admin ◆ Server filename = <code><install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml</code>. 4. Verify that the business process runs successfully <p>Debug information for this service can be found in the FTP Client adapter and services log files.</p>
Notes	<p>Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:</p> <ul style="list-style-type: none"> ◆ FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. ◆ FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command.

Implementing the FTP Client SITE Service

To implement the FTP Client SITE service, complete the following tasks:

1. Create an FTP Client SITE service configuration (or enable the configuration installed with Application and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client SITE service. For information, see *Configuring the FTP Client SITE Service* on page 537.

3. Use the FTP Client SITE service in a business process.

Configuring the FTP Client SITE Service

To configure the FTP Client SITE service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
ResponseTimeout	Maximum number of seconds it can take for the trading partner system to respond before the service times out and terminates. If you specify a number less than 30, 30 seconds is used. Optional. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SiteCommand	A site-specific control command. Required. Valid value is any site-specific command that does not require the use of a data connection.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Output from Service to Business Process

The following table contains the parameters passed from the FTP Client SITE service to the business process:

Parameter	Description
ServerResponse	Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required.
TranscriptDocumentId	Identifies the document that contains a transcript of the exact exchange with the FTP server. Required.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client SITE service:

Parameter	Description
ResponseTimeout	Maximum number of seconds it can take for the trading partner system to respond before the service times out and terminates. If you specify a number less than 30, 30 seconds is used. Optional. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter.
SessionToken	Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. Note: The session token is returned from the FTP Client Begin Session service.
SiteCommand	A site-specific control command. Required. Valid value is any site-specific command that does not require the use of a data connection.
SaveTranscript	Indicates how to handle the transcript. Valid values are: <ul style="list-style-type: none">◆ erroronly – persists the transcript only when an error occurs◆ on – always persists the transcript Default is on. Optional.

Business Process Example

The following example business process illustrates using the FTP Client SITE service:

```
<sequence>
  [[ Insert FTP Client Begin Session ]]
  <operation name="FTP SITE SERVICE">
    <participant name="FTPClientSite"/>
    <output message="SiteRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="SiteCommand">Help</assign>
    </output>
    <input message="inmsg">
      <assign to="FTPClientSiteResults" from="*"></assign>
    </input>
  </operation>
  [[ Insert FTP Client End Session ]]
</sequence>
```

FTP Reverse Proxy Adapter

The FTP Reverse Proxy adapter provides a high level of data protection between external connections using FTP and your company's Application server. Use this adapter to receive documents from a trading partner who uses the FTP protocol.

The FTP Reverse Proxy adapter can only be used with the Sterling Secure Proxy 2.0 (or later) product. See the Sterling Secure Proxy documentation for information.

FTP Server Adapter (Build 4300 - Build 4308)

The following table provides an overview of the FTP Server adapter:

System name	FTP Server Adapter
Graphical Process Modeler (GPM) category	None
Description	This adapter receives and processes requests from external trading partners that are submitted using the FTP protocol. This adapter is used with a Application Perimeter server.
Business usage	Use this adapter to put files into a Application mailbox or get files from a Application mailbox.
Usage example	A trading partner uses an FTP client to retrieve a business document from his Application mailbox. The FTP Server adapter receives and processes the trading partner request.
Preconfigured?	A configuration of the FTP Server adapter is installed with Application, but is disabled by default. You can enable the preconfigured FTP Server adapter, or create a new configuration from Application.
Requires third party files?	Certicom SSL Library (currently available in Application)
Platform availability	All supported Application platforms
Related services	None
Application requirements	To log in to the FTP Server you must have permission to your virtual root (either explicitly assigned or defaulted). To access a mailbox, you must have permission to that mailbox and all mailboxes that may be between it and your virtual root. If users exceed a maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	The FTP Server adapter does not directly initiate business processes. However, mailbox activities can trigger routing rules.
Invocation	Not used in business processes
Business process context considerations	None
Returned status values	None

Restrictions	<ul style="list-style-type: none"> ◆ FTP Server is tightly integrated with the Application mailbox system. An FTP client can only access the mailbox that is assigned to its user account. ◆ FTP Server does not support all functions specified in RFC 0959 (Standard FTP Server). It supports basic functions to integrate with the Application mailbox system such as list message and sub-mailbox, send and extract message to/from mailbox. ◆ FTP Server is not integrated with business process invocation when processing a request from a client. ◆ The home directory for FTP is a virtual root mailbox in Application. Mailboxes include both extractable and nonextractable messages. When accessing a mailbox using the FTP Server adapter, only extractable messages are displayed. To change this default behavior, edit the ftpserver.properties file and set <i>listUnextractables=true</i> (default is <i>false</i>). ◆ The timeout value for a control channel connection is controlled by a parameter in the ftpserver.properties file. The default timeout value is 600 seconds. The minimum value is 60 seconds. If the control channel is idle longer than the timeout value, the session is terminated, unless the data channel is open (whether or not data is being transferred). ◆ To access the FTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root. ◆ Restricted operation can be granted to users with a parameter named <i>MailboxLoginWithoutVirtualRootPermission</i>. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes.
Persistence level	None. This adapter does not have a pre-set persistence level.
Testing considerations	<p>At Application startup, attempt to access the FTP Server using a supported FTP client with the configured IP address and port.</p> <p>Debug information can be found in the FTP logs. Select Logging Level from the following:</p> <ul style="list-style-type: none"> ◆ Error - only errors. ◆ Communication Trace - errors, requests from clients, and responses from the Server adapter. This includes ACL violations. ◆ All - for debugging, all activities.

Implementing the FTP Server Adapter

To implement the FTP Server adapter, complete the following tasks:

1. Create an FTP Server adapter configuration (or enable the configuration installed with Application and edit parameters as needed).
2. Configure the FTP Server adapter.

Configuring the FTP Server Adapter

To configure the FTP Server adapter, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Not applicable for this adapter. Leave at default.
FTP Server Listen Port	The port number that the FTP Server should bind to and listen on for connection requests. The default value depends on the system platform and on your Application configuration. Required.
Active Data Port Range	<p>A range of ports that the server can allocate for the transfer of data to or from the FTP client in active mode. Optional. Example values are:</p> <ul style="list-style-type: none">◆ 1024-2048◆ 2222◆ 3000-4000 <p>Note: You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799.</p> <p>If left blank, the server will choose available system ports.</p>
Passive Data Port Range	<p>A range of ports that the server can allocate for the transfer of data to or from the FTP client in passive mode. Optional. Example values are:</p> <ul style="list-style-type: none">◆ 1024-2048◆ 2222◆ 3000-4000 <p>Note: You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799.</p> <p>If left blank, the server will choose available system ports.</p>
Perimeter Server	<p>Select a Perimeter server from the list. Default is node1 & local. Required.</p> <p>Note: You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring.</p>
Transfer Buffer Size (bytes)	Specifies the size in bytes of the buffer used when transferring a file. Required. Valid values are 0 to 9,999,999,999. Default is 32000.
Minimum Number of Threads	<p>A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be less than or equal to the Maximum Number of Threads value. Default is 3. Required.</p> <p>Note: Retain the default value unless instructed otherwise by Sterling Commerce support.</p>

Field	Description
Maximum Number of Threads	<p>A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be greater than or equal to the Minimum Number of Threads value. Default is 6. Required.</p> <p>Note: Retain the default value unless instructed otherwise by Sterling Commerce support.</p>
NAT Address	<p>Specifies the NAT IP address that the FTP server should send to the user FTP client in the passive connection mode. Optional. Overrides the global NAT address specified in the ftpserver.properties file.</p>
Maximum Logins	<p>Maximum number of logins the adapter may have active at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 999999999.</p>
Maximum Logins per user	<p>Maximum number of logins each user may have active on this adapter at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 999999999.</p>
Document Storage	<p>Indicates whether the body of the request document must be stored on the file system or if it should be in the database. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default – If your system administrator has changed the installed default of File System, this ensures that the correct location is used. ◆ Database – body of the request document will be stored in the database ◆ File System (default) – this is the default value when Application is installed, but it can be changed. Contact your system administrator to see if the default has been changed. <p>Required.</p> <p>Note: For more information about document storage types, see <i>Managing Services and Adapters</i>.</p>
Should the adapter be restricted to a certain group of users?	<p>Select Yes or No to indicate whether to restrict specific users and groups to access the FTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages.</p>
Should the restricted users be assigned a specific range of ports?	<p>Select Yes or No to indicate whether to assign a specific port, range, or ranges of ports to the users. Required. Default is No. If Yes, specify <i>User Active Ports</i>, <i>User Passive Ports</i>, <i>Group Active Ports</i>, and or <i>Group Passive Ports</i> on subsequent pages. You can specify any or all of these fields.</p>
Users	<p>Select a list of users who are granted permission to access the server.</p>
Groups	<p>Select a list of groups who are granted permission to access the server.</p>
<i>User Active Ports</i>	<p>Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>

Field	Description
User Passive Ports	<p>Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Group Active Ports	<p>Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Group Passive Ports	<p>Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Extractable Count	<p>The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer. Optional.</p>
Extractable For	<p>Indicates the length of time (in days, hours and minutes) the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable Count. Valid value is in the format <i>dddhhmm</i>. Optional.</p>
Extractable	<p>Whether the message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Valid values are Yes and No. Optional. Default is Yes.</p>
SSL	<p>Whether Secure Sockets Layer (SSL) is active. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – If SSL is requested by a client it will be rejected. (default) ◆ Optional – SSL is used if requested by a client. ◆ Must – Clients that do not request SSL are not allowed to authenticate. <p>Note: If Optional or Must is specified, the asset protection key must enable SSL for the appropriate protocol.</p>
Key Certificate Passphrase	<p>Password that protects the server key certificate. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional.</p>

Field	Description
Cipher Strength	Strength of the algorithms used to encrypt data. Valid values are: <ul style="list-style-type: none"> ◆ ALL ◆ WEAK – Often required for international e-commerce, because government regulations prohibit STRONG encryption from being exported. ◆ STRONG Default is STRONG. Required if SSL option is Must or Optional.
Key Certificate (System Store)	Private key and certificate for server authentication. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional.
CA Certificates	Certificate used to validate the certificate of an FTP client. This is the public key. If no CA certificate is chosen, no client certification is performed. Optional.
Clear Command Channel	Indicates that communication across the command channel is not encrypted after authentication is completed. Optional.

FTP Server Functions Supported

The following table contains the FTP functions that are supported with the FTP Server adapter:

Category	Commands Supported
Access Control commands	<ul style="list-style-type: none"> ◆ USER – User name ◆ PASS – Password ◆ CWD – Change Working Directory ◆ CDUP – Change to Parent Directory ◆ QUIT – Logout
Transfer Parameter Commands	<ul style="list-style-type: none"> ◆ PORT – Data port ◆ PASV – Passive mode ◆ TYPE – Representation type (ASCII and Binary) ◆ STRU – File Structure (File) ◆ MODE – Transfer mode (Stream)

Category	Commands Supported
Service Commands	<ul style="list-style-type: none"> ◆ DELE – Delete ◆ RETR – Retrieve ◆ STOR – Store ◆ ABOR – Abort ◆ PWD – Print Working Directory ◆ XPWD – Print Working Directory (legacy format) ◆ LIST – List ◆ NLST – Name List ◆ HELP – Help ◆ NOOP – No Operation ◆ RNFR – Rename From ◆ RNTO – Rename To ◆ SITE – Site Parameter (CPWD and HELP) ◆ SYST – System
Security Commands	<ul style="list-style-type: none"> ◆ AUTH – Authentication/Security Mechanism ◆ CCC – Clear Command Channel ◆ PBSZ – Protect Buffer Size ◆ PROT – Data Channel Protection Level

FTP Server Functions Not Supported

The following table contains the FTP functions that are not supported with the FTP Server adapter:

Category	Commands Not Supported
Access Control commands	<ul style="list-style-type: none"> ◆ ACCT – Account ◆ SMNT – Structure Mount ◆ REIN – Re-initialize
Transfer Parameter Commands	<ul style="list-style-type: none"> ◆ TYPE – Representation type (EBCDIC and Local Byte) ◆ STRU – File Structure (Record and Page) ◆ MODE – Transfer mode (Block and Compressed)

Category	Commands Not Supported
Service Commands	<ul style="list-style-type: none">◆ STOU – Store Unique◆ APPE – Append◆ ALLO – Allocate◆ REST – Restart◆ RMD – Remove Directory◆ MKD – Make Directory◆ STAT – Status

Activity Types for the FTP Server Adapter

This adapter reports the following activities to the Services Controller for activity monitoring:

PUT – Adds a file to a mailbox

GET – Retrieves a file from a mailbox

Session – Records all activity after connection

FTP Server Adapter (Build 4309 - Build 4323)

The following table provides an overview of the FTP Server adapter:

System name	FTP Server Adapter
Graphical Process Modeler (GPM) category	None
Description	This adapter receives and processes requests from external trading partners that are submitted using the FTP protocol. This adapter is used with a Application Perimeter server.
Business usage	Use this adapter to put files into a Application mailbox or get files from a Application mailbox.
Usage example	A trading partner uses an FTP client to retrieve a business document from his Application mailbox. The FTP Server adapter receives and processes the trading partner request.
Preconfigured?	A configuration of the FTP Server adapter is installed with Application, but is disabled by default. You can enable the preconfigured FTP Server adapter, or create a new configuration from Application.
Requires third party files?	Certicom SSL Library (currently available in Application)
Platform availability	All supported Application platforms
Related services	None
Application requirements	To log in to the FTP Server you must have permission to your virtual root (either explicitly assigned or defaulted). To access a mailbox, you must have permission to that mailbox and all mailboxes that may be between it and your virtual root. If users exceed a maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	The FTP Server adapter does not directly initiate business processes. However, mailbox activities can trigger routing rules.
Invocation	Not used in business processes
Business process context considerations	None
Returned status values	None

Restrictions	<ul style="list-style-type: none"> ◆ FTP Server is tightly integrated with the Application mailbox system. An FTP client can only access the mailbox that is assigned to its user account. ◆ FTP Server does not support all functions specified in RFC 0959 (Standard FTP Server). It supports basic functions to integrate with the Application mailbox system such as list message and sub-mailbox, send and extract message to/from mailbox. ◆ FTP Server is not integrated with business process invocation when processing a request from a client. ◆ The home directory for FTP is a virtual root mailbox in Application. Mailboxes include both extractable and nonextractable messages. When accessing a mailbox using the FTP Server adapter, only extractable messages are displayed. To change this default behavior, edit the ftpserver.properties file and set <i>listUnextractables=true</i> (default is <i>false</i>). ◆ The timeout value for a control channel connection is controlled by a parameter in the ftpserver.properties file. The default timeout value is 600 seconds. The minimum value is 60 seconds. If the control channel is idle longer than the timeout value, the session is terminated, unless the data channel is open (whether or not data is being transferred). ◆ To access the FTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root. ◆ Restricted operation can be granted to users with a parameter named <i>MailboxLoginWithoutVirtualRootPermission</i>. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes.
Persistence level	None. This adapter does not have a pre-set persistence level.
Testing considerations	<p>At Application startup, attempt to access the FTP Server using a supported FTP client with the configured IP address and port.</p> <p>Debug information can be found in the FTP logs. Select Logging Level from the following:</p> <ul style="list-style-type: none"> ◆ Error - only errors. ◆ Communication Trace - errors, requests from clients, and responses from the Server adapter. This includes ACL violations. ◆ All - for debugging, all activities.

Implementing the FTP Server Adapter

To implement the FTP Server adapter, complete the following tasks:

1. Create an FTP Server adapter configuration (or enable the configuration installed with Application and edit parameters as needed).
2. Configure the FTP Server adapter.

Configuring the FTP Server Adapter

To configure the FTP Server adapter, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Not applicable for this adapter. Leave at default.
FTP Server Listen Port	The port number that the FTP Server should bind to and listen on for connection requests. The default value depends on the system platform and on your Application configuration. Required.
Active Data Port Range	<p>A range of ports that the server can allocate for the transfer of data to or from the FTP client in active mode. Optional. Example values are:</p> <ul style="list-style-type: none">◆ 1024-2048◆ 2222◆ 3000-4000 <p>Note: You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799.</p> <p>If left blank, the server will choose available system ports.</p>
Passive Data Port Range	<p>A range of ports that the server can allocate for the transfer of data to or from the FTP client in passive mode. Optional. Example values are:</p> <ul style="list-style-type: none">◆ 1024-2048◆ 2222◆ 3000-4000 <p>Note: You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799.</p> <p>If left blank, the server will choose available system ports.</p>
Perimeter Server	<p>Select a Perimeter server from the list. Default is node1 & local. Required.</p> <p>Note: You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring.</p>
Transfer Buffer Size (bytes)	Specifies the size in bytes of the buffer used when transferring a file. Required. Valid values are 0 to 9,999,999,999. Default is 32000.
Minimum Number of Threads	<p>A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be less than or equal to the Maximum Number of Threads value. Default is 3. Required.</p> <p>Note: Retain the default value unless instructed otherwise by Sterling Commerce support.</p>

Field	Description
Maximum Number of Threads	<p>A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be greater than or equal to the Minimum Number of Threads value. Default is 6. Required.</p> <p>Note: Retain the default value unless instructed otherwise by Sterling Commerce support.</p>
NAT Address	<p>Specifies the NAT IP address that the FTP server should send to the user FTP client in the passive connection mode. Optional. Overrides the global NAT address specified in the ftpserver.properties file.</p>
Maximum Logins	<p>Maximum number of logins the adapter may have active at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 999999999.</p>
Maximum Logins per user	<p>Maximum number of logins each user may have active on this adapter at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 999999999.</p>
Document Storage	<p>Indicates whether the body of the request document must be stored on the file system or if it should be in the database. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default – If your system administrator has changed the installed default of File System, this ensures that the correct location is used. ◆ Database – body of the request document will be stored in the database ◆ File System (default) – this is the default value when Application is installed, but it can be changed. Contact your system administrator to see if the default has been changed. <p>Required.</p> <p>Note: For more information about document storage types, see <i>Managing Services and Adapters</i>.</p>
Should the adapter be restricted to a certain group of users?	<p>Select Yes or No to indicate whether to restrict specific users and groups to access the FTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages.</p>
Should the restricted users be assigned a specific range of ports?	<p>Select Yes or No to indicate whether to assign a specific port, range, or ranges of ports to the users. Required. Default is No. If Yes, specify <i>User Active Ports</i>, <i>User Passive Ports</i>, <i>Group Active Ports</i>, and or <i>Group Passive Ports</i> on subsequent pages. You can specify any or all of these fields.</p>
Users	<p>Select a list of users who are granted permission to access the server.</p>
Groups	<p>Select a list of groups who are granted permission to access the server.</p>
<i>User Active Ports</i>	<p>Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>

Field	Description
User Passive Ports	<p>Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Group Active Ports	<p>Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Group Passive Ports	<p>Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Extractable Count	<p>The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer. Optional.</p>
Extractable For	<p>Indicates the length of time (in days, hours and minutes) the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable Count. Valid value is in the format <i>dddhhmm</i>. Optional.</p>
Extractable	<p>Whether the message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Valid values are Yes and No. Optional. Default is Yes.</p>
SSL	<p>Whether Secure Sockets Layer (SSL) is active. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – If SSL is requested by a client it will be rejected. (default) ◆ Optional – SSL is used if requested by a client. ◆ Must – Clients that do not request SSL are not allowed to authenticate. <p>Note: If Optional or Must is specified, the asset protection key must enable SSL for the appropriate protocol.</p>
Key Certificate Passphrase	<p>Password that protects the server key certificate. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional.</p>

Field	Description
Cipher Strength	Strength of the algorithms used to encrypt data. Valid values are: <ul style="list-style-type: none"> ◆ ALL ◆ WEAK – Often required for international e-commerce, because government regulations prohibit STRONG encryption from being exported. ◆ STRONG Default is STRONG. Required if SSL option is Must or Optional.
Key Certificate (System Store)	Private key and certificate for server authentication. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional.
CA Certificates	Certificate used to validate the certificate of an FTP client. This is the public key. If no CA certificate is chosen, no client certification is performed. Optional.
Clear Command Channel	Indicates that communication across the command channel is not encrypted after authentication is completed. Optional.

FTP Server Functions Supported

The following table contains the FTP functions that are supported with the FTP Server adapter:

Category	Commands Supported
Access Control commands	<ul style="list-style-type: none"> ◆ USER – User name ◆ PASS – Password ◆ CWD – Change Working Directory ◆ CDUP – Change to Parent Directory ◆ QUIT – Logout
Transfer Parameter Commands	<ul style="list-style-type: none"> ◆ PORT – Data port ◆ PASV – Passive mode ◆ TYPE – Representation type (ASCII and Binary) ◆ STRU – File Structure (File) ◆ MODE – Transfer mode (Stream)

Category	Commands Supported
Service Commands	<ul style="list-style-type: none"> ◆ DELE – Delete ◆ RETR – Retrieve ◆ STOR – Store ◆ ABOR – Abort ◆ PWD – Print Working Directory ◆ XPWD – Print Working Directory (legacy format) ◆ LIST – List ◆ NLST – Name List ◆ HELP – Help ◆ NOOP – No Operation ◆ RNFR – Rename From ◆ RNTO – Rename To ◆ SITE – Site Parameter (CPWD and HELP) ◆ SYST – System
Security Commands	<ul style="list-style-type: none"> ◆ AUTH – Authentication/Security Mechanism ◆ CCC – Clear Command Channel ◆ PBSZ – Protect Buffer Size ◆ PROT – Data Channel Protection Level ◆ REST – Restart

FTP Server Functions Not Supported

The following table contains the FTP functions that are not supported with the FTP Server adapter:

Category	Commands Not Supported
Access Control commands	<ul style="list-style-type: none"> ◆ ACCT – Account ◆ SMNT – Structure Mount ◆ REIN – Re-initialize
Transfer Parameter Commands	<ul style="list-style-type: none"> ◆ TYPE – Representation type (EBCDIC and Local Byte) ◆ STRU – File Structure (Record and Page) ◆ MODE – Transfer mode (Block and Compressed)

Category	Commands Not Supported
Service Commands	<ul style="list-style-type: none">◆ STOU – Store Unique◆ APPE – Append◆ ALLO – Allocate◆ RMD – Remove Directory◆ MKD – Make Directory◆ STAT – Status

Activity Types for the FTP Server Adapter

This adapter reports the following activities to the Services Controller for activity monitoring:

PUT – Adds a file to a mailbox

GET – Retrieves a file from a mailbox

Session – Records all activity after connection

FTP Server Adapter (Build 4324 or higher)

The following table provides an overview of the FTP Server adapter:

System name	FTP Server Adapter
Graphical Process Modeler (GPM) category	None
Description	This adapter receives and processes requests from external trading partners that are submitted using the FTP protocol. This adapter is used with a Application Perimeter server.
Business usage	Use this adapter to put files into a Application mailbox or get files from a Application mailbox.
Usage example	A trading partner uses an FTP client to retrieve a business document from his Application mailbox. The FTP Server adapter receives and processes the trading partner request.
Preconfigured?	A configuration of the FTP Server adapter is installed with Application, but is disabled by default. You can enable the preconfigured FTP Server adapter, or create a new configuration from Application.
Requires third party files?	Certicom SSL Library (currently available in Application)
Platform availability	All supported Application platforms
Related services	None
Application requirements	To log in to the FTP Server you must have permission to your virtual root (either explicitly assigned or defaulted). To access a mailbox, you must have permission to that mailbox and all mailboxes that may be between it and your virtual root. If users exceed a maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	The FTP Server adapter does not directly initiate business processes. However, mailbox activities can trigger routing rules.
Invocation	Not used in business processes
Business process context considerations	None
Returned status values	None

Restrictions	<ul style="list-style-type: none"> ◆ FTP Server is tightly integrated with the Application mailbox system. An FTP client can only access the mailbox that is assigned to its user account. ◆ FTP Server does not support all functions specified in RFC 0959 (Standard FTP Server). It supports basic functions to integrate with the Application mailbox system such as list message and sub-mailbox, send and extract message to/from mailbox. ◆ FTP Server is not integrated with business process invocation when processing a request from a client. ◆ The home directory for FTP is a virtual root mailbox in Application. Mailboxes include both extractable and nonextractable messages. When accessing a mailbox using the FTP Server adapter, only extractable messages are displayed. To change this default behavior, edit the ftpserver.properties file and set <i>listUnextractables=true</i> (default is <i>false</i>). ◆ The timeout value for a control channel connection is controlled by a parameter in the ftpserver.properties file. The default timeout value is 600 seconds. The minimum value is 60 seconds. If the control channel is idle longer than the timeout value, the session is terminated, unless the data channel is open (whether or not data is being transferred). ◆ To access the FTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root. ◆ Restricted operation can be granted to users with a parameter named <i>MailboxLoginWithoutVirtualRootPermission</i>. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes.
Persistence level	None. This adapter does not have a pre-set persistence level.
Testing considerations	<p>At Application startup, attempt to access the FTP Server using a supported FTP client with the configured IP address and port.</p> <p>Debug information can be found in the FTP logs. Select Logging Level from the following:</p> <ul style="list-style-type: none"> ◆ Error - only errors. ◆ Communication Trace - errors, requests from clients, and responses from the Server adapter. This includes ACL violations. ◆ All - for debugging, all activities.

Implementing the FTP Server Adapter

To implement the FTP Server adapter, complete the following tasks:

1. Create an FTP Server adapter configuration (or enable the configuration installed with Application and edit parameters as needed).
2. Configure the FTP Server adapter.

Configuring the FTP Server Adapter

To configure the FTP Server adapter, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Not applicable for this adapter. Leave at default.
FTP Server Listen Port	The port number that the FTP Server should bind to and listen on for connection requests. The default value depends on the system platform and on your Application configuration. Required.
Active Data Port Range	<p>A range of ports that the server can allocate for the transfer of data to or from the FTP client in active mode. Optional. Example values are:</p> <ul style="list-style-type: none">◆ 1024-2048◆ 2222◆ 3000-4000 <p>Note: You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799.</p> <p>If left blank, the server will choose available system ports.</p>
Passive Data Port Range	<p>A range of ports that the server can allocate for the transfer of data to or from the FTP client in passive mode. Optional. Example values are:</p> <ul style="list-style-type: none">◆ 1024-2048◆ 2222◆ 3000-4000 <p>Note: You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799.</p> <p>If left blank, the server will choose available system ports.</p>
Perimeter Server	<p>Select a Perimeter server from the list. Default is node1 & local. Required.</p> <p>Note: You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring.</p>
Transfer Buffer Size (bytes)	Specifies the size in bytes of the buffer used when transferring a file. Required. Valid values are 0 to 9,999,999,999. Default is 32000.
Minimum Number of Threads	<p>A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be less than or equal to the Maximum Number of Threads value. Default is 3. Required.</p> <p>Note: Retain the default value unless instructed otherwise by Sterling Commerce support.</p>

Field	Description
Maximum Number of Threads	<p>A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be greater than or equal to the Minimum Number of Threads value. Default is 6. Required.</p> <p>Note: Retain the default value unless instructed otherwise by Sterling Commerce support.</p>
NAT Address	<p>Specifies the NAT IP address that the FTP server should send to the user FTP client in the passive connection mode. Optional. Overrides the global NAT address specified in the ftpserver.properties file.</p>
Maximum Logins	<p>Maximum number of logins the adapter may have active at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 999999999.</p>
Maximum Logins per user	<p>Maximum number of logins each user may have active on this adapter at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 999999999.</p>
Document Storage	<p>Indicates whether the body of the request document must be stored on the file system or if it should be in the database. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default – If your system administrator has changed the installed default of File System, this ensures that the correct location is used. ◆ Database – body of the request document will be stored in the database ◆ File System (default) – this is the default value when Application is installed, but it can be changed. Contact your system administrator to see if the default has been changed. <p>Required.</p> <p>Note: For more information about document storage types, see <i>Managing Services and Adapters</i>.</p>
Should the adapter be restricted to a certain group of users?	<p>Select Yes or No to indicate whether to restrict specific users and groups to access the FTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages.</p>
Should the restricted users be assigned a specific range of ports?	<p>Select Yes or No to indicate whether to assign a specific port, range, or ranges of ports to the users. Required. Default is No. If Yes, specify <i>User Active Ports</i>, <i>User Passive Ports</i>, <i>Group Active Ports</i>, and or <i>Group Passive Ports</i> on subsequent pages. You can specify any or all of these fields.</p>
Users	<p>Select a list of users who are granted permission to access the server.</p>
Groups	<p>Select a list of groups who are granted permission to access the server.</p>
<i>User Active Ports</i>	<p>Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>

Field	Description
User Passive Ports	<p>Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Group Active Ports	<p>Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Group Passive Ports	<p>Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:</p> <ul style="list-style-type: none"> ◆ 3000 ◆ 4000-5000, 6000 <p>Optional.</p>
Extractable Count	<p>The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer. Optional.</p>
Extractable For	<p>Indicates the length of time (in days, hours and minutes) the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable Count. Valid value is in the format <i>dddhhmm</i>. Optional.</p>
Extractable	<p>Whether the message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Valid values are Yes and No. Optional. Default is Yes.</p>
SSL	<p>Whether Secure Sockets Layer (SSL) is active. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – If SSL is requested by a client it will be rejected. (default) ◆ Optional – SSL is used if requested by a client. ◆ Must – Clients that do not request SSL are not allowed to authenticate. <p>Note: If Optional or Must is specified, the asset protection key must enable SSL for the appropriate protocol.</p>
Key Certificate Passphrase	<p>Password that protects the server key certificate. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional.</p>

Field	Description
Cipher Strength	Strength of the algorithms used to encrypt data. Valid values are: <ul style="list-style-type: none"> ◆ ALL ◆ WEAK – Often required for international e-commerce, because government regulations prohibit STRONG encryption from being exported. ◆ STRONG Default is STRONG. Required if SSL option is Must or Optional.
Key Certificate (System Store)	Private key and certificate for server authentication. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional.
CA Certificates	Certificate used to validate the certificate of an FTP client. This is the public key. If no CA certificate is chosen, no client certification is performed. Optional.
Clear Command Channel	Indicates that communication across the command channel is not encrypted after authentication is completed. Optional.

FTP Server Functions Supported

The following table contains the FTP functions that are supported with the FTP Server adapter:

Category	Commands Supported
Access Control commands	<ul style="list-style-type: none"> ◆ USER – User name ◆ PASS – Password ◆ CWD – Change Working Directory ◆ CDUP – Change to Parent Directory ◆ QUIT – Logout
Transfer Parameter Commands	<ul style="list-style-type: none"> ◆ PORT – Data port ◆ PASV – Passive mode ◆ TYPE – Representation type (ASCII and Binary) ◆ STRU – File Structure (File) ◆ MODE – Transfer mode (Stream)

Category	Commands Supported
Service Commands	<ul style="list-style-type: none"> ◆ DELE – Delete ◆ RETR – Retrieve ◆ STOR – Store ◆ ABOR – Abort ◆ PWD – Print Working Directory ◆ XPWD – Print Working Directory (legacy format) ◆ LIST – List ◆ NLST – Name List ◆ HELP – Help ◆ NOOP – No Operation ◆ RNFR – Rename From ◆ RNTO – Rename To ◆ SITE – Site Parameter (CPWD and HELP) ◆ SYST – System
Security Commands	<ul style="list-style-type: none"> ◆ AUTH – Authentication/Security Mechanism ◆ CCC – Clear Command Channel ◆ PBSZ – Protect Buffer Size ◆ PROT – Data Channel Protection Level ◆ REST – Restart

FTP Server Functions Not Supported

The following table contains the FTP functions that are not supported with the FTP Server adapter:

Category	Commands Not Supported
Access Control commands	<ul style="list-style-type: none"> ◆ ACCT – Account ◆ SMNT – Structure Mount ◆ REIN – Re-initialize
Transfer Parameter Commands	<ul style="list-style-type: none"> ◆ TYPE – Representation type (EBCDIC and Local Byte) ◆ STRU – File Structure (Record and Page) ◆ MODE – Transfer mode (Block and Compressed)

Category	Commands Not Supported
Service Commands	<ul style="list-style-type: none">◆ STOU – Store Unique◆ APPE – Append◆ ALLO – Allocate◆ RMD – Remove Directory◆ MKD – Make Directory◆ STAT – Status

Activity Types for the FTP Server Adapter

This adapter reports the following activities to the Services Controller for activity monitoring:

PUT – Adds a file to a mailbox

GET – Retrieves a file from a mailbox

Session – Records all activity after connection

GS:Unix Purge Process Service

The following table provides an overview of the GS:Unix (Gentran:Server[®] for UNIX[®]) Purge Process service:

System name	None
Graphical Process Modeler (GPM) category	None
Description	Deletes lifecycle records created before a date and time designated by you.
Preconfigured?	No
Requires third party files?	LC221 and LCDESTINFO tables created in Gentran:Server for UNIX.
Platform availability	<ul style="list-style-type: none">◆ Sun Solaris◆ HP-UX◆ IBM-AIX
Related services	No
Application requirements	Gentran:Server for UNIX
Initiates business processes?	No
Invocation	Runs by the Application scheduler.
Business process context considerations	No

How the GS:Unix Purge Process Service Works

The GS:Unix Purge Process service deletes records from the LIFECYCLE and LIFECYCLE_EXTENSION tables created in Application and from existing LC221 and LCDESTINFO tables created in Gentran:Server for UNIX. The service runs as a step in a business process or by itself on a schedule.

For example, consider the following scenario. Today is August 12, 2004 and the time is 3 p.m. You want to delete files older than three days and five hours. Using the GS:Unix Purge Process service, you can delete files created before August 9, 2004 at 10 a.m.

Implementing the GS:Unix Purge Process Service


To implement the GS:Unix Purge Process service for use in a business process, complete the following tasks:

1. Activate your license for the GS:Unix Purge Process service. See *An Overview of Implementing Services*.
2. Create a GS:Unix Purge Process service configuration. See *Creating a Service Configuration*.
3. Configure the GS:Unix Purge Process service. See *Configuring the GS:Unix Purge Process Service* on page 565.
4. Use the GS:Unix Purge Process service in a business process.

Configuring the GS:Unix Purge Process Service

To configure the GS:Unix Purge Process service, you must specify settings for the following fields in Application:

Note: The names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: See <i>Using Service Groups</i> .
Run As User	Applies to the scheduling of the business process. The Run As User field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes. Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list. Valid values: Any valid Application user ID Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.
Use 24 Hour Clock Display	If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.

Field	Description
Schedule	<p>Information about scheduling the business process after the File System adapter collects files. The Schedule field only displays as an option if <i>Start a business process once files are collected</i> is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If you select this field, the adapter does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.
Lifespan Days (lifespandays)	Number of days from the current day used to determine which records to delete from the LIFECYCLE, LIFECYCLE_EXTENSION, LC221, and LCDESTINFO tables. Optional
Lifespan Hours (lifespanhours)	Number of hours from the current time used to determine which records to delete from the LIFECYCLE, LIFECYCLE_EXTENSION, LC221, and LCDESTINFO tables. Optional.
Business Process (initialWorkflowName)	Business process where the records are located. Optional.

Gentran:Server Windows Adapter

The following table provides an overview of the Gentran:Server® Windows® adapter:

System name	GenSrvNT
Graphical Process Modeler (GPM) category	All Services and Applications Sterling > Gentran
Description	Enables you to transfer data between Application and Gentran:Server for Windows as part of a business process within Application.
Preconfigured?	No
Requires third party files?	<ul style="list-style-type: none">◆ An installed and configured copy of Gentran:Server for Windows version 3.1.1 or higher◆ The host name of the computer where Gentran:Server for Windows is installed◆ The name of the Gentran:Server for Windows mailbox that this adapter configuration will use to exchange data with Application <p>Note: The Gentran:Server Windows adapter.</p> <ul style="list-style-type: none">◆ adapter client .jar file installed, configured, and running
Platform availability	<ul style="list-style-type: none">◆ Microsoft Windows◆ Sun Solaris◆ HP-UX◆ IBM-AIX◆ United Linux◆ RedHat AS
Related services	No
Application requirements	No
Initiates business processes?	Yes
Invocation	Runs by a scheduler or another adapter in a business process.

Requirements

To configure and use the Gentran:Server Windows adapter, you must be familiar with Gentran:Server for Windows and must have:

- An installed and configured copy of Gentran:Server for Windows version 3.1.1 or higher
- The host name of the computer where Gentran:Server for Windows is installed

The name of the Gentran:Server for Windows mailbox that this adapter configuration will use to exchange data with Application

The Gentran:Server Windows adapter client .jar file installed, configured, and running

How the Gentran:Server Windows Adapter Works

The Gentran:Server Windows adapter is used in a business process to exchange data between Application and Gentran:Server for Windows.

The following sections summarize how the Gentran:Server Windows adapter works in a business process.

Application Receives Data from Gentran:Server for Windows

The following procedure illustrates how Application receives data from Gentran:Server for Windows:

1. Using the scheduling parameters set for the adapter in Application, the Gentran:Server Windows adapter polls Gentran:Server for Windows to check if it has data in the Application mailbox.
2. If data is in the mailbox, the Gentran:Server Windows adapter retrieves the data through the remote RMI service of the adapter and the Application gateway.
3. When Application receives the data, the Gentran:Server Windows adapter starts the business process defined in the adapter configuration.

Application Sends Data to Gentran:Server for Windows

The following procedure illustrates how Application sends data to Gentran:Server for Windows:

1. The Gentran:Server Windows adapter can be called in a business process to send data to Gentran:Server for Windows.
2. Using the Gentran:Server Windows adapter remote RMI service and Application, the data is sent to Gentran:Server for Windows.
3. The Application gateway creates the appropriate message and sends the data to the receiver.

Implementing the Gentran:Server Windows Adapter

To implement the Gentran:Server Windows adapter, complete the following tasks:

1. Activate your license for the Gentran:Server Windows adapter. See *An Overview of Implementing Services*.
2. Install the Gentran:Server Windows adapter. See *Installing the Adapter Client on Gentran:Server for Windows* on page 570.
3. Install the adapter client on Gentran:Server for Windows. See *Installing the Adapter Client on Gentran:Server for Windows* on page 570.
4. Install Application Gateway on Gentran:Server for Windows. See *Installing Application Gateway in Gentran:Server for Windows* on page 570.
5. Create a Application Mailbox on Gentran:Server for Windows. See *Creating a Application Mailbox in Gentran:Server for Windows* on page 571.

6. Create a Gentran:Server Windows adapter configuration. See *Creating a Service Configuration*.
7. Configure the Gentran:Server Windows adapter. See *Configuring the Gentran:Server Windows Adapter* on page 571.
8. Use the configured services in a business process.

Installing Java on Gentran:Server for Windows

Application installs a copy of the appropriate Java Development Kit (JDK) required for proper operation. You must ensure that there is a JDK installed on the machine where Gentran:Server for Windows is installed and that it matches the same JDK version as the one installed with Application (wherever it is installed). If Application is installed on the same machine as Gentran:Server for Windows then the correct JDK version will automatically be installed when Application is installed. The system requirements for this installation are Java Runtime Environment/Java Virtual Machine version 1.3.

Installing the Adapter Client on Gentran:Server for Windows

To install the Gentran:Server Windows adapter client on Gentran:Server for Windows:

1. In the `<install_dir>/installed_data/gentrannt/client` folder on the computer where Application is installed, locate the `GSNTClient.jar` file.
2. Move a copy of the `GSNTClient.jar` file to the `GentranInstallation/bin` folder on the computer where Gentran:Server for Windows is installed.
3. Unpack the `GSNTClient.jar` file in the `GentranInstallation/bin` folder. `GSNTClient.jar` contains the following files:
 - ◆ `GenSrvNtAdapter.jar` (Note: do not unpack the `GenSrvNtAdapter.jar` file!)
 - ◆ `runremotermi.bat`
 - ◆ `SterlingIntegrator<ver>.dll`
 - ◆ `Manifest.mf`
4. Rename the appropriate version of `SterlingIntegrator<ver>.dll` to `SterlingIntegrator.dll`. Note that the version of `SterlingIntegrator.dll` corresponds with the version number of Gentran:Server for Windows.

Installing Application Gateway in Gentran:Server for Windows

To install a Application gateway in Gentran:Server for Windows:

1. Run **regedit** on the computer where Gentran:Server for Windows is installed and navigate to:
`HKEY_LOCAL_MACHINE\Software\Sterling
Commerce\Gentran\Mailbox\version#\Client\Gateways`
2. Right-click the **Client\Gateways** folder and select **New > String Value**.
3. In the **String Value Name** box, type **STERLINGIntegrator** as the name of the gateway.
4. In the **Data** box, type **SterlingIntegrator.dll** to assign the file to the Application gateway.
The *xx* in `SterlingIntegratorxx.dll` relates to the version number of Gentran:Server for Windows that you have installed on your computer.

5. Navigate to:
HKEY_LOCAL_MACHINE\Software\Sterling
Commerce\Gentran\Mailbox\version#\Server\Gateways
6. Right-click the **Server\Gateways** folder and select **New > String Value**.
7. In the **String Value Name** box, type **STERLINGIntegrator** as the name of the gateway.
8. In the **Data** box, type **SterlingIntegrator.dll** to assign the file to the Application gateway.
9. Start the Mailbox service to establish the new gateway settings.

Creating a Application Mailbox in Gentran:Server for Windows

The SterlingIntegrator.dll file must be placed in the registry before you can create the Application mailbox using the Application gateway.

To create a Application mailbox in Gentran:Server for Windows:

1. In Gentran:Server for Windows, start MailBox Server Manager.
2. Right-click **Mailboxes** and select **Create**.
3. Type a unique name as the name of the mailbox you are creating and click **Next** twice.
4. Click **Yes, use this mailbox as a gateway**, select Application as the type of gateway to use with this mailbox, and then click **Next**.
5. On the summary page, confirm that the information you entered is correct, and then click **Finish**.
Depending on your system configuration, the mailbox properties window may open.
6. Open the Gentran:Server Mailbox Manager client, right-click **Client\Gateway**, and select **Properties**.
7. Type the content type and subtype of messages and attachments when receiving messages.
8. Click **Edit Recipients** and select the receivers to send the message to.

Using Remote RMI in Gentran:Server for Windows

To use remote RMI in Gentran:Server for Windows:

1. In the runremotermi.bat file, note the port number at the end of the last line in the file. This is the port number required in the RMI server address during configuration of the Gentran:Server Windows adapter.
If this port number changes, you must change the configuration of the Gentran:Server for Windows adapter.
2. Run the runremotermi.bat file.

Configuring the Gentran:Server Windows Adapter

To configure the Gentran:Server Windows adapter, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the Gentran:Server Windows adapter in Application:

Note: The names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
Host Name (gentranhost)	Name of the computer where Gentran:Server for Windows is installed.
Mailbox Name (remoteMailbox)	<p>Name of the mailbox created in Gentran:Server for Windows that this configuration of the adapter will use.</p> <p>Note: This mailbox must have Application listed as its gateway.</p>
RMI Server Address (remoteAddr)	Address of the RMI server that is installed on the same computer as Gentran:Server for Windows. The format to enter this address is <i>ipadr:port/rmiServerName</i> .
Start a Business Process When Data is Received? (bootstrap)	Whether the adapter should start a business process. Valid values are Yes and No. Required.
Business process (initialWorkFlowId)	Business process to start when the Gentran:Server Windows adapter receives data. Required only if you selected Yes in the Start a Business Process When Data is Received? field.
Do not use schedule	If this field is selected, this service does not start a business process and does not run on a schedule. Required for receiving data from Application.
Run service based on timer every	Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup.
Run service daily at	Valid values are the hour and minutes at which to run the service, daily. Indicate whether you want the service to run at startup.
Run service weekly on	Valid values are the day of the week, the hour, and the minutes at which to run the service. Indicate whether you want the service to run at startup.

GPM Configuration

The following table describes the fields used to configure the Gentran:Server Windows adapter in the GPM:

Field	Description
Config	Name of the adapter configuration.
Action	Action that this configuration of the adapter performs. Valid values are Receive from Gentran and Send to Gentran.

Get Document Info Service

The Get Document Info service provides information about the primary document. You can query the service about many types of information, including DocumentName, DocumentBodyLength, and DocumentID. The following table provides an overview of the Get Document Info service:

System name	GetDocumentInfoServiceType
Graphical Process Modeler (GPM) category	All Services
Description	<p>The Get Document Info service provides information about the primary document. You can query the service about the following information:</p> <ul style="list-style-type: none">◆ DocumentName◆ DocumentBodyLength◆ DocumentId◆ DocumentSubject◆ DocumentCreateTime◆ DocumentContentType◆ DocumentCharEncoding◆ DocumentContentSubType◆ DocumentPreviousDocId◆ DocumentLength◆ Correlations◆ Storage Type◆ Node name◆ File name◆ Storage time <p>You can also use the service to set the following information, with or without creating a new document in the system:</p> <ul style="list-style-type: none">◆ BodyName◆ Document Content Type◆ DocumentCharEncoding◆ DocumentContentSubType◆ updateMetaDataOnly: When set to “true” BodyName, Document Content Type, DocumentCharEncoding, DocumentContentSub Type, Document Subject, DocumentName, and DocumentSize can be updated without creating a new document.

Business Usage	This service allows information obtained from the document to be used in the business process at run time. You can also use the service to set certain information about the document in the business process.
Usage Example	Obtain correlations from a document, such as a correlation for document type, and use this to determine further processing that would be specific to that document type
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates Business Processes	This service does not initiate a business process
Invocation	Not applicable
Business Process Considerations	Allows a document ID to be passed instead of the whole document. This makes information from the document available for use in the process without opening the document.
Returned Status Values	<ul style="list-style-type: none"> ◆ Success ◆ Error
Restrictions	None
Persistence Level	System default

Implementing the Get Document Info Service

To implement the Get Document Info service, complete the following tasks:

1. Create a Get Document Info service configuration. See *Creating a Service Configuration*.
2. Configure the Get Document Info service. See *Configuring the Get Document Info Service* on page 576.
3. Use the Get Document Info service in a business process.

Configuring the Get Document Info Service

To configure the Get Document Info service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration. Required.
BodyName	Body Name of the document. Optional.
CORRELATION	Whether to pass correlations associated with the document. Optional. Valid values: <ul style="list-style-type: none">◆ true – Pass document correlation parameters◆ false – Do not pass document correlation parameters Default is false.
DOCUMENT_ID	Document ID to look up from the workflow context. If there is no document ID, it will look up the primary document instead. Optional. Valid value: <i>hostname:e52fd2:fd70c53c8c:-7648</i> . Note: When you specify DOCUMENT_ID, the document specified is loaded into the Primary Document. If a Primary Document already exists, it will be overwritten.
DocumentCharEncoding	Type of character encoding used in the document. Optional. Example: iso-8859-1.
DocumentContentSubType	Content subtype of the document. Optional. Example: plain.
DocumentContentType	Content type of the document. Optional. Example: text.
Encrypted	Indicates if the document is encrypted. Optional.

Once configured, these parameters are passed from the business process to the Get Document Info service.

Parameters Passed from Business Process to Service

The following parameters are passed from the Business Process to the Get Document Info service:

Field	Description
DocumentId	Document ID to look up from the workflow context. If there is no document ID, it will look up the primary document instead. Optional. Example: <i>hostname:e52fd2:fd70c53c8c:-7648</i> .
DocumentContentType	The current document content type. Optional.
DocumentContentSubType	The current document content sub-type. Optional.
DocumentCharEncoding	The current document character encoding. Optional
Correlation	The current document correlation. Optional.
BodyName	The current document body name. Optional
doClone	Set to true to modify the document metadata in the clone or copy of the existing document. The modified data is in the cloned document. Valid values are true and false. Optional. For example, when you alter the value in DocumentContentType, setting doClone to true creates a clone of the existing document but with the modified value. If you do not set the doClone parameter or set it to false, the existing document will have the new metadata value and the modified value is available in the process data.

Field	Description
updateMetaDataOnly	<p>A flag that indicates if the document data is for a metadata update only. Optional. Valid values are: True or False. If set to true, the service can set the following metadata without creating a new document that is stored in the database:</p> <ul style="list-style-type: none"> ◆ doc name ◆ bodyName ◆ char_encoding ◆ subject ◆ content_type ◆ content_subtype ◆ doc size <p>Default is false.</p> <p>Example of a document with updateMetaDataOnly set to true:</p> <pre> <operation name="SetContentType"> <participant name="GetDocumentInfoService"/> <output message="xout"> <assign to="." from="*" /> <assign to="DocumentContentType">text</assign> <assign to="DocumentContentSubType">plain</assign> <assign to="DocumentCharEncoding">iso-8859-1</assign> <assign to="updateMetaDataOnly">>true</assign> </output> <input message="xin"> <assign to="." from="*" /> </input> </operation> </pre> <p>Example of a document that is not using updateMetaDataOnly and is set to false, or not set at all. A new document will be saved in the database:</p> <pre> <operation name="SetContentType"> <participant name="GetDocumentInfoService"/> <output message="xout"> <assign to="." from="*" /> <assign to="DocumentContentType">text</assign> <assign to="DocumentContentSubType">plain</assign> <assign to="DocumentCharEncoding">iso-8859-1</assign> <assign to="updateMetaDataOnly">>false</assign> </output> <input message="xin"> <assign to="." from="*" /> </input> </operatiom> </pre>

Parameters Passed from Service to Business Process

The following parameters are passed from the Get Document Info service to the business process:

Field	Description
PrimaryDocument	The link of the primary document. Optional. Example: <i>hostname:73ce5c:ff993119fc:-733f</i>
BodyName	The current body name. Required.
BodyLength	The current body length. Required. Default is 0.
DocumentBodyLength	The current document body length. Required. Default is 0.
DocumentName	The current document name. Optional.
DocumentId	The current document ID. Required. Example: <i>hostname:331059:ff998a9313:-7f71</i>
DocumentSubject	The current document subject. Required.
DocumentCreateTime	The current document creation time. Required.
DocumentContentType	The current document content type. Optional.
DocumentCharEncoding	The current document character encoding. Optional.
DocumentContentSubType	The current document content sub-type. Optional.
DocumentPreviousDocId	The previous document ID. Optional.
DocumentLength	The current document length. Optional.
DocumentLifeSpan	The current document life span. Optional.
DocumentInitialId	The current document initial ID. Optional.
DocumentMaxInlineBodySize	The current document maximum inline body size. Optional.
DocumentPurgeAfter	The current document purge after. Optional. Valid values: any timestamp value.
DocumentStorageType	The current document storage type. Optional.
DocumentTrackingId	The current document tracking ID. Optional.
DocumentWorkflowId	The current document workflow ID. Optional.
DocumentFileName	The current document file name. Optional.
DocumentNodeName	The current document node name. Optional.
DocumentType	The current document type. Optional.
Encrypted	Indicates if the document is encrypted. Optional.
UpdateMetaDataOnly	Indicates if the document data is for a metadata update only. Optional.

Business Process Examples

Example 1

Example of the getdocinfo service that shows processing of the primary document created by the xmlencoder.

```
<process name="GetDocumentInfoService_01">
  <sequence name="simple">
    <operation name="Set Document">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">process_data_to_document</assign>
        <assign to="root_element">Document1</assign>
        <assign to="XPath">/ProcessData</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>

    <operation name="SetContentType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="." from="*" />
        <assign to="DocumentContentType">text</assign>
        <assign to="DocumentContentSubType">plain</assign>
        <assign to="DocumentCharEncoding">iso-8859-1</assign>
      </output>
      <input message="xin">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Example 2

Example of a process document that is collected by the filesystemadapter. The second getinfoservice is processing a document by a document ID.

```
<process name="GetDocumentInfoService_02">
  <sequence name="simple">
    <!-- Create File System adapter instances for input and output -->
    <!-- Place the document you want processed in the input instance -->
    <operation name="FileSystemAdapter">
      <participant name="FileSystemAdapter"/>
      <output message="outputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="filter">My.log</assign>
        <assign to="useSubFolders">>false</assign>
        <assign to="docStorageType">fs</assign>
        <assign to="bootstrap">>false</assign>
        <assign to="deleteAfterCollect">>false</assign>
        <assign to="." from="*"></assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

    <input message="inputMessage">
      <assign to="." from="*"></assign>
    </input>
  </operation>
<operation name="FileSystemAdapter">
  <participant name="FileSystemAdapter"/>
  <output message="outputMessage">
    <assign to="Action">FS_EXTRACT</assign>
    <assign to="useSubFolders">>false</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inputMessage">
    <assign to="." from="*"></assign>
  </input>
</operation>
<operation name="GetDocumentInfoService">
  <participant name="GetDocumentInfoService"/>
  <output message="xout">
    <assign to="." from="*" />
    <assign to="DocumentContentType">text</assign>
    <assign to="DocumentContentSubType">plain</assign>
    <assign to="DocumentCharEncoding">iso-8859-1</assign>
  </output>
  <input message="xin">
    <assign to="." from="*" />
  </input>
</operation>
<operation name="GetDocumentInfoService">
  <participant name="GetDocumentInfoService"/>
  <output message="xout">
    <assign to="." from="*" />
    <
    <!-- The DOCUMENT_ID parameter must contain a correct value from -->
    <!-- the Document table in the business process database.-->
    <assign to="DOCUMENT_ID">somehost:e61fd1:fd70c58c8c:-7478</assign>
    <assign to="DocumentContentType">text</assign>
    <assign to="DocumentContentSubType">plain</assign>
    <assign to="DocumentCharEncoding">iso-8859-1</assign>
  </output>
  <input message="xin">
    <assign to="." from="*" />
  </input>
</operation>
</sequence>
</process>

```

Example 3

Example of the correlation for the primary document that is created by the translator.

```

<process name = "GetDocumentInfoService_04">
  <!-- GetDocumentInfoService_04 map must be checked in -->
  <!-- This business process uses GetDocumentInfoService_04_input_data.xml -->
  <rule name="New">
    <condition>WF_CORRELATIONS/correlation/value/text() = 'N'</condition>
  </rule>

  <rule name="Not New">

```

```

    <condition>WF_CORRELATIONS/correlation/value/text() != 'N'</condition>
</rule>

<sequence>
  <sequence name="Translate and Report">
    <operation name="Translation">
      <participant name="Translation"/>
      <output message="TranslationTypeInputMessage">
        <assign to="map_name">PurchaseOrder_Metadata_Test</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="Correlation Service">
      <participant name="CorrelationService"/>
      <output message="CorrelationServiceTypeInputMessage">
        <assign to="NAME">BPMETA_MESSAGE_TYPE</assign>
        <assign to="VALUE">PurchaseOrder</assign>
        <assign to="TYPE">BUSINESS_PROCESS</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="GetDocumentInfoService">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="CORRELATION">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
<choice name="Check POType">
  <select>
    <case ref="New" activity="ProcessNewPO"/>
    <case ref="Not New" activity="ProcessOtherPO"/>
  </select>
</choice>

```

GXS ICS FTP Adapter

The following table provides an overview of the GXS (Global Exchange Services) ICS (Interchange Services) FTP adapter:

System name	GXS ICS FTP Adapter
Graphical Process Modeler (GPM) category	None
Description	The GXS ICS FTP Adapter is used to send and collect documents from the GXS Interchange Services (ICS) mailbox via the TCP/IP FTP Gateway.
Business usage	This adapter can be used to exchange documents with the Trading Partner that uses the GXS Interchange Services mailbox.
Usage example	An internal service invokes a business process that sends a trading document to the trading partner mailbox in the GXS VAN. The BPML can be invoked manually or by an internal service.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	FTP Client Adapter
Application requirements	None
Initiates business processes?	Yes, this adapter initiates business processes when it is in the Collect mode.
Invocation	This adapter is invoked by an internal service.
Business process context considerations	The configuration parameters and the outgoing document are picked up by the GXS ICS FTP adapter in the workflow context.
Workflow Context Considerations	You must be familiar with the internal service (workflow context/business process parameters) that invoked this adapter. Workflow parameters are passed into the internal service and business process parameters are the values specified in the business process code.
Returned status values	<ul style="list-style-type: none">◆ 0 – Success◆ 1 – Error
Restrictions	None
Persistence level	System default

Testing considerations To test an instance of the GXS ICS FTP adapter, create a GXS_ICS_Adapter by clicking **Deployment->Services->Configuration** and choosing **GXS ICS FTP Adapter**. The most frequent problems encountered are:

- ◆ Incorrect parameters are set up.
- ◆ The adapter is not active.

If you get an error with any of these conditions, check with your System Administrator.

Implementing the GXS ICS FTP Adapter

To implement the GXS ICS FTP adapter, complete the following tasks:

1. Activate your license for the GXS ICS FTP adapter. See *An Overview of Implementing Services*.
2. Create an GXS ICS FTP adapter configuration. See *Creating a Service Configuration*.
3. Configure the adapter. See *Configuring the GXS ICS FTP Adapter* on page 584.
4. Create and enable a business process that includes the GXS ICS FTP adapter.
5. Test the business process and the adapter.
6. Run the business process.

Configuring the GXS ICS FTP Adapter


To configure the GXS ICS FTP adapter, you must specify field settings in Application. See *Creating a Service Configuration*.

The following table describes the fields used to configure the GXS ICS FTP adapter in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: See <i>Using Service Groups</i> .

Field	Description
Default GXS ICS FTP Server	DNS name or IP address of the GXS ICS FTP server that Application is to connect to. This is any valid DNS name or a dotted decimal address. The default value (myhost) is shown as an example. Required
Default GXS ICS FTP Server Port	Port number of the GXS ICS FTP server. This is typically port number 21, but it can be another number depending on the FTP server settings. Default is 21. Required.
User Name	User login name to connect to the GXS ICS FTP server. Alphanumeric and case-sensitive. Required.
Password	Password used for authentication when connecting to the GXS ICS FTP server. Case-sensitive. Required. The password will be obscured (encrypted) and will appear as all asterisks.
SMIME Encryption User Certificate	Not used
SMIME Decryption Certificate	Not used
SSL	Flag that determines Secure Socket Layering (SSL) socket negotiation. Required. Valid values: <ul style="list-style-type: none"> ◆ SSL_IMPLICIT – Indicates that the GXS ICS FTP server expects and requires SSL to happen automatically at the time of connection. ◆ SSL_EXPLICIT – Indicates that the GXS ICS FTP client requests SSL and a secure connection is negotiated. ◆ None – Indicates that the connection will not use SSL.
Clear Control Channel	Indicates if information that travels across the control channel should be clear. Required. Valid values: <ul style="list-style-type: none"> ◆ Yes – Information will be clear ◆ No – Information does not need to be clear
Key Certificate Passphrase	Not used
Cipher Strength	Level of encryption to be applied to the data that flows through the socket connection. Optional. Valid values: <ul style="list-style-type: none"> ◆ ALL – All cipher strengths are supported ◆ WEAK – Often required for international trade, because government regulations prohibit STRONG encryption from being exported. ◆ STRONG – Highest level of encryption Default is STRONG.

Field	Description
Key Certificate (System Store)	PrivateKeys / Public Certificates that are signed by the Trading Partner Trusted Certificate Authority. In process data, this parameter is displayed as an object ID. Select a valid key certificate from the list of those already stored inside Application. Required if the GXS ICS FTP server requires client authentication using SSL.
CA Certificates	Trusted Certificate Authority public certificate. In process data, this parameter is displayed as an object ID. Select a valid CA certificate from the list of those already stored inside Application. Required for server authentication using SSL.
Type of operation	Type of operation to perform Valid values: <ul style="list-style-type: none"> ◆ Collect – Collect messages from the GXS ICS mailbox ◆ Submit – Submit mailbox messages to the GXS ICS mailbox
Select the type of message to collect	Type of message to collect from the GXS ICS mailbox. Required if <i>Type of operation</i> is set to Collect. Valid values: <ul style="list-style-type: none"> ◆ Mailbox messages – Collect mailbox messages. ◆ Report – Retrieve a report. Default is Mailbox messages.
Select the report you want to retrieve	Type of report to retrieve. Required if <i>Type of operation</i> is set to Collect and <i>Select the type of message to collect</i> is set to Report. Valid values: <ul style="list-style-type: none"> ◆ Sender Status Report ◆ Sender Summary ◆ Sender Unretrieved ◆ Sender Error ◆ Receiver Status ◆ Receiver Summary ◆ Third Party Reject ◆ Trading Pair Trxn Detail
Specify the message retrieval criteria	Whether to collect all messages or only those meeting the criteria specified in <i>Sender or Receiver address</i> or in <i>Match this ILog number</i> . Required if <i>Type of operation</i> is set to Collect and <i>Select the type of message to collect</i> is set to Mailbox messages. Valid values: <ul style="list-style-type: none"> ◆ Download all interchanges ◆ Download interchanges that match the criteria specify below Default is Download all interchanges.
Sender or Receiver address	Address on the GXS ICS FTP server to download messages from. Optional.
Match this ILog number	ILog number to download messages from the GXS ICS FTP server under. Optional.

Field	Description
Specify the message download option	<p>How to save the downloaded documents. Required if <i>Type of operation</i> is set to Collect and <i>Select the type of message to collect</i> is set to Mailbox messages.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Download all interchanges into a single document ◆ Download each interchanges as an individual document <p>Default value is Download all interchanges into a single document</p>
Business Process	<p>Name of the business process you want this adapter to start. Required if <i>Type of operation</i> is set to Collect.</p>
Run As User	<p>Applies to the scheduling of the business process. The Run As User field only displays as an option if <i>Type of operation</i> is set to Collect.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid values:</p> <p>Any valid Application user ID</p> <p>Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>
Use 24 Hour Clock Display	<p>If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.</p>

Field	Description
Schedule	<p>Information about scheduling the collection action. Required if <i>Type of operation</i> is set to Collect.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.

Before Using the GXS ICS FTP Adapter in a Business Process

If the GXS ICS FTP server that is communicating with Application uses SSL communication, the CA certificate object ID or system key certificate object ID (CACert or Keycert) files must be saved before the GXS ICS FTP adapter can be used in a business process. The Application system administrator must save these certificates and private key in the Application database.

Business Process Configuration

No GXS ICS FTP adapter parameters are configured with the Graphical Process Modeler (GPM). All parameters must be configured in Application.

ILOG Number

A unique ILOG number is generated per file upload to the GXS ICS FTP server. The adapter captures the ILOG number and stores it in the process data. This enables the downstream workflow to act on to this information. The ILOG number is stored in the following format:

```
<ProcessData>
  <GXSResponses>
    <IlogNumber>7311258167</IlogNumber>
  </GXSResponses>
</ProcessData>
```

Certificate Authority (CA) Certificate

Currently, a GXS Server accepts any valid Verisign class 3 CA certificate. You are required to import these certificates into Application (using **Trading Partners > Digital Certificates > CA**). One of these imported certificates must be selected when you configure the GXS ICS FTP adapter instance. Customers are encouraged to check with GXS to ensure that the GXS Server is still using Verisign as its CA prior to using the GXS ICS FTP adapter.

Example Business Process

The following sample BPML invokes an instance of the GXS ICS FTP adapter.

```
<process name="Test_bpml">
  <sequence name="optional">
    <operation name="Invoke an instance of GXS ICS FTP adapter ">
      <participant name="GXS_ICS_FTP_adapter"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

HTTP Client Adapter

The following table provides an overview of the HTTP Client adapter:

System name	HTTP Client Adapter
Graphical Process Modeler (GPM) category	None
Description	<p>Sends HTTP requests to trading partners using a Perimeter server.</p> <p>The HTTP Client adapter replaces the B2B HTTP Client adapter and the HTTP Send adapter, which are being retired. The HTTP Client adapter and its related services provide all the functionality of the former two adapters plus the following enhancements:</p> <ul style="list-style-type: none">◆ Use Perimeter services◆ Highly scalable (>150 concurrent transfers)◆ Supports large files up to 2 GB◆ Supports HTTP 1.1
Business usage	Use this adapter to send documents to a trading partner using HTTP.
Usage example	The Application business process must send a message to a trading partner and the trading partner's profile specifies HTTP as the transport protocol. The business process passes the document as well as information about the trading partner to the HTTP Client adapter. Using the information provided in the business process, the HTTP Client adapter connects to the trading partner and transfers the document, enforcing any security specifications from the business process.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ HTTP Client Begin Session Service◆ HTTP Client End Session Service◆ HTTP Client GET Service◆ HTTP Client Method Service◆ HTTP Client POST Service
Application requirements	<p>An HTTP server is required at the external trading partner location.</p> <p>When this adapter is configured with a non-local-mode Perimeter server, the Perimeter server must be installed and running. This Perimeter server is typically installed in a DMZ environment, separated from Application by a firewall.</p>
Initiates business processes?	No

Invocation	The HTTP Client adapter is invoked from one of the HTTP Client services used within a business process.
Business process context considerations	Business processes using HTTP client services should not be marked as Auto Resume. These services require an established session which will no longer exist after a restart.
Returned status values	<ul style="list-style-type: none"> ◆ 100s – Informational parameters and more status parameters will follow. ◆ 200s – Request was OK. ◆ 300s – Request failed and more status codes will follow. ◆ 400s – Client Request Failed. This is a client error. ◆ 500s – Server failed to handle a valid request. This is a server error.
Restrictions	All work bound within an HTTP Client Begin Session service and an HTTP Client End Session service must be within the same business process.
Persistence level	N/A
Testing considerations	<p>To test this adapter, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the <i>Business Process Example</i>.</p> <p>Debug information related to this adapter can be found in the HTTP Client Adapter and Services log.</p>

Implementing the HTTP Client Adapter

To implement the HTTP Client adapter, complete the following tasks:

1. Create an HTTP Client adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client adapter. For information, see *Configuring the HTTP Client Adapter*.

Configuring the HTTP Client Adapter

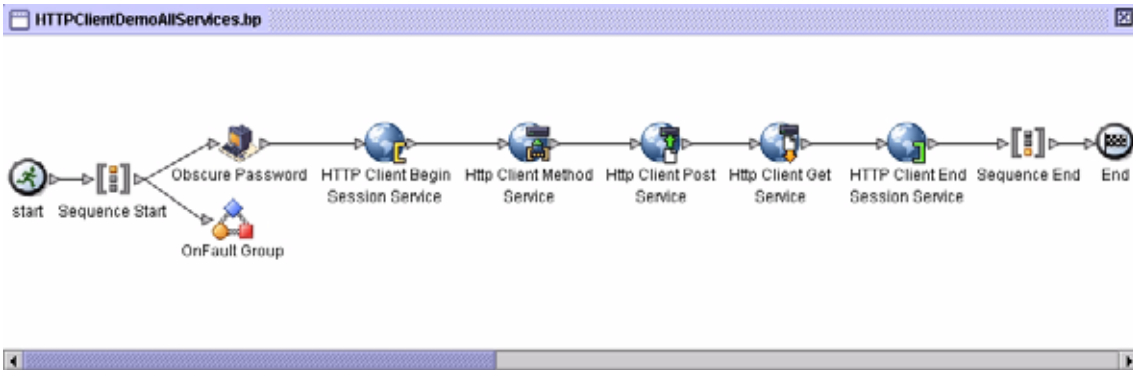
To configure the HTTP Client adapter, you must specify field settings in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Perimeter Server Name	List of Perimeter servers, including local-mode Perimeter servers. Required.
Local port range	Ranges or lists of local ports to restrict the adapter to. For example, 123-456 or 123,124,999.
Number of connection retries	Number of times the HTTP Client adapter will try to connect to the server. Required. Valid value is any integer between 0 and 50. Default is 3.
Delay between retries (second)	Number of seconds the HTTP Client adapter will wait between retry attempts. Required. Valid value is any integer between 1 and 7200. Default is 20.
Response timeout (second)	Number of seconds it should take for the server to respond. Required. Valid value is any integer between 1 and 999999. Default is 60.
Use proxy server	<p>Whether to use an HTTP proxy server. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – A proxy server will be used ◆ No – No proxy server will be used
Proxy hostname	IP address or host name of the proxy server to use. Required if Use proxy server is set to Yes. This parameter can be overridden by a Trading Partner profile setting.
Proxy port	Port number of the proxy server. Required if Use proxy server is set to Yes. This parameter can be overridden by a Trading Partner profile setting.
Proxy retry attempts	Number of times the HTTP Client adapter will try to connect to the proxy server. Required if Use proxy server is set to Yes. Valid values: any integer between 0 and 50. Default is 3. This parameter can be overridden by a Trading Partner profile setting.
Use Proxy Basic Authentication	<p>Whether the proxy server requires basic authentication. Required if Use proxy server is set to Yes.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Proxy basic authentication will be used ◆ No – Proxy basic authentication will not be used
Proxy username	User name to use with the proxy server. Required if Use Proxy Basic Authentication is set to Yes.
Proxy password	Password for the Proxy username. Required if User Proxy Basic Authentication is set to Yes.

Business Process Example

The following example business process illustrates the use of the HTTP Client services, which work through the HTTP Client adapter:



The associated BPML code is shown below:

```
<process name="HTTPClientDemoAllServices">
  <sequence>
    <!-- Get obscured password -->
    <operation name="Obscure Password">
      <participant name="HTTPClientObscureParameter"/>
      <output message="outmsg">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientObscureResults" from="*"></assign>
      </input>
    </operation>

    <!-- HTTP Client Begin Session service-->
    <!-- Create connection to specified host and port -->
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">httpserver</assign>
        <assign to="RemotePort">12345</assign>

        <!-- If server requires Basic authentication -->
        <assign to="RemoteUserId">userid</assign>
        <assign to="RemotePasswd"
from="HTTPClientObscureResults/admin/text()"></assign>
RemoteUserId and RemotePasswd are not needed if using a trading partner profile. The
profile will supply this information.

        <!-- To use a Trading Partner profile -->
        <assign to="ProfileId">someExistingProfileId</assign>

        <!-- To override HTTPClientAdapter configuration settings -->
```

```

    <assign to="ConnectionRetries">10</assign>
    <assign to="RetryDelay">1</assign>

    <!-- If server requires SSL authentication -->
    <assign to="SSL">Must</assign>
    <assign to="CipherStrength">Strong</assign>
    <assign to="CACertificateId">SomeHttp-Id</assign>
    <assign to="SystemCertificateId">httpsampleclientcert1-Id</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
  </input>
</operation>

  <!-- HTTP Client Method Service -->
  <operation name="HTTP Client Method Service">
    <participant name="HTTPClientMethod"/>
    <output message="HTTPClientMethodServiceTypeInputMessage">
      <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="MethodType">HEAD</assign>
      <assign to="URI">/hello</assign>
      <assign to="RawResponse">true</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="HTTPClientMethodServiceResults" from="*" append="true"/>
    </input>
  </operation>

  <!-- HTTP Client POST Service -->
  <operation name="HTTP Client Post Service">
    <participant name="HTTPClientPost"/>
    <output message="HTTPClientPostServiceTypeInputMessage">
      <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="URI">/hello</assign>
      <assign to="RawResponse">false</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="HTTPClientPostServiceResults" from="*" append="true"/>
    </input>
  </operation>

  <!-- HTTP Client GET Service -->
  <operation name="HTTP Client Get Service">
    <participant name="HTTPClientGet"/>
    <output message="HTTPClientGetServiceTypeInputMessage">
      <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="URI">/someURI/</assign>
      <assign to="ResponseTimeout">120</assign>
      <assign to="." from="*"></assign>
    </output>
  </operation>

```

```

    </output>
    <input message="inmsg">
      <assign to="HTTPClientGetServiceResults" from="*" append="true"/>
    </input>
  </operation>

  <!-- HTTP Client End Session Service -->
  <!-- Ends session specified by SessionToken -->
  <operation name="HTTP Client End Session Service">
    <participant name="HTTPClientEndSession"/>
    <output message="HTTPClientEndSessionServiceTypeInputMessage">
      <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="HTTPClientEndSessionServiceResults" from="*"
append="true"/>
    </input>
  </operation>

  <!-- Provides error handling -->
  <onFault>
    <sequence name="End Session">
      <operation name="HTTP Client End Session Service">
        <participant name="HTTPClientEndSession"/>
        <output message="HTTPClientEndSessionServiceTypeInputMessage">
          <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
        </input>
      </operation>
    </sequence>
  </onFault>

</sequence>
</process>

```

HTTP Client Begin Session Service

The following table provides an overview of the HTTP Client Begin Session service:

System name	HTTP Client Begin Session Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > HTTP Client
Description	The HTTP Client Begin Session service is used to start an HTTP session with an external trading partner for the purpose of exchanging business documents. This service works through an instance of the HTTP Client adapter.
Business usage	Use this service to establish a session with a trading partner HTTP server.
Usage examples	A business process is executed that translates a document that must be sent to a trading partner. After the translation, the application looks up information about how to transport data to the trading partner identified in the trading partner profile. The trading partner profile specifies HTTP as the transport protocol. The application then uses the HTTP Client Begin Session service to establish a session with the trading partner's HTTP server.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported application platforms
Related services	<ul style="list-style-type: none">◆ HTTP Client adapter◆ HTTP Client End Session service◆ HTTP Client GET service◆ HTTP Client Method service◆ HTTP Client POST service <p>To mask the values associated with the remote password parameter, use the Obscure Data - Process Data Values service in conjunction with the HTTP Begin Session service. This service is presented in the GPM as Obscure Parameter on the All Services stencil.</p>
Application requirements	An HTTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	The HTTP Client Begin Session service allows you to specify a remote password. In order for this password to be obscured in process data for the business process, you should also use the Obscure Data - Process Data Values service within the same business process. The Obscure Data - Process Data Values service may be used to mask the values associated with parameters.

Returned status values	0 – Success 1– Error
Restrictions	None
Persistence level	System Default
Testing considerations	To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the HTTP Client adapter. Debug information for this service can be found in the HTTP Client adapter and services log files.

Implementing the HTTP Client Begin Session Service

To implement the HTTP Client Begin Session service, complete the following tasks:

1. Create an HTTP Client Begin Session service configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client Begin Session service. For information, see *Configuring the HTTP Client Begin Session Service*.
3. Use the HTTP Client Begin Session service in a business process.

Configuring the HTTP Client Begin Session Service

You can set the following values in the trading partner profile and specify it in the ProfileId field, or you can set these values in an instance of the service to only apply for that instance. If specified in the HTTP Client Begin Session service, the following values override those in the HTTP trading partner profile:

- ◆ CACertificateId
- ◆ CipherStrength
- ◆ ConnectionRetries
- ◆ RemoteHost
- ◆ RemotePasswd
- ◆ RemotePort
- ◆ RemoteUserId
- ◆ SSL
- ◆ SystemCertificateId
- ◆ RetryDelay

To configure the HTTP Client Begin Session service, you must specify settings for the following fields in the GPM:

Field	Description
Name	Name this service will have in the application
Description	Description of service
Select a Group	<p>Select one of the following options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Config	Name of the service configuration.
CACertificateId	<p>List of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID. Required if SSL is set to Must.</p> <p>Obtain an SSL certificate from a Certificate Authority or from your trading partner. Check it into the application from the Admin menu, selecting Trading Partner > Digital Certificates > CA to make it available in this list.</p>
CipherStrength	<p>The level of encryption to apply to the data that flows through the socket connection. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ ALL – WEAK or STRONG is accepted ◆ WEAK – 40 bit encryption is required ◆ STRONG – 128 bit or higher encryption is required (default)
ConnectionRetries	<p>The number of times the service will try to connect to the Trading Partner Systems. Connection retries occur only with TCP/IP related issues. Optional. Valid value is any numeric value.</p> <p>Note: The value entered for this parameter overrides the Number of connection retries setting in the HTTP Client adapter configuration.</p>
DelayWaitingOnIO	<p>The amount of time (in seconds) that a business process using the Http Client Begin Service adapter waits before going to WAITING_ON_IO state and frees up the engine resources for other processes. Optional. Valid value is an integer.</p> <p>If you specify a positive integer, the parameter specifies the number of seconds the business process has to wait for a response from the HTTP server before going to WAITING_ON_IO state.</p> <p>If you specify a negative integer, the business process waits for the response from the HTTP server to complete. The business process does not go to WAITING_ON_IO state.</p> <p>If you specify 0, the business process goes to WAITING_ON_IO state after sending a request to the HTTP server.</p> <p>If you specify a value lesser than -1, the parameter value is set to 0 (default value).</p> <p>Note: The value you specify in the httpclient.properties file for the defaultDelayWaitingOnIO property overrides the setting you specified in the GPM.</p>

Field	Description
HTTPClientAdapter	Select the HTTP Client adapter for this service to use when beginning a session with an HTTP server. Required.
ProfileId	Trading partner profile identification. Optional. Valid value is any valid profile ID.
RemoteHost	External Trading Partner host system (HTTP server IP Address or DNS name). Required. Use any valid IP Address or DNS name.
RemotePasswd	HTTP remote login password. Optional. Note: For the password to be masked in process data, the Obscure Data - Process Data Values service must also be used in the same business process. The name used to store the password must be the same as the specified RemoteUserId.
RemotePort	External Trading Partner port number. Required.
RemoteUserId	HTTP remote login username. Optional.
RetryDelay	Number of seconds the adapter will wait before retrying. Optional. Valid value is any numeric value. Note: The value entered for this parameter overrides the Delay between retries setting in the HTTP Client adapter configuration.
SSL	Determines SSL socket negotiation. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Must – SSL socket negotiation is enabled. ◆ None – Connection will not use SSL. Default.
SystemCertificateId	Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. This certificate confirms the identity of the client to the server. Required if SSL is set to Must and the server requires client authentication. Obtain the certificate from your trading partner. Check it into the application from the Admin menu, selecting Trading Partner > Digital Certificates > System to make it available in this list.

Output from Service to Business Process

The following table contains the parameters passed from the HTTP Client Begin Session service to the business process:

Parameter	Description
SessionToken	Specifies the identifier for the session established between the HTTP Client adapter and an HTTP server. Required.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client Begin Session service:

Field	Description
CACertificateId	Drop-down menu that contains a list of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID. Required if SSL is set to Must.
CipherStrength	The level of encryption to apply to the data that flows through the socket connection. Valid values are: <ul style="list-style-type: none">◆ ALL – WEAK or STRONG is accepted◆ WEAK – 40 bit encryption is required◆ STRONG – 128 bit or higher encryption is required
HTTPClientAdapter	Select the HTTP Client adapter for this service to use when beginning sessions with an HTTP server. Required.
ConnectionRetries	The number of times the service will try to connect to the Trading Partner Systems. Connection retries occur only with TCP/IP related issues. Optional. Valid value is any numeric value. Default is 1.
ProfileId	Trading partner profile identification. Optional. Valid value is any valid profile ID.
RemoteHost	External Trading Partner host system (HTTP server IP Address or DNS name. Required. Use any valid IP Address or DNS name.
RemotePasswd	HTTP remote login password. Optional. Note: The password will be obscured using the Obscure service.
RemotePort	External Trading Partner port number. Required.
RemoteUserId	HTTP remote login username. Optional.
RetryDelay	Number of seconds the adapter will wait before retrying. Optional. Valid value is any numeric value. Default is 1.
SessionBeginTime	Specifies the date and time that the session started. Required.
SSL	Determines SSL socket negotiation. Valid values are: <ul style="list-style-type: none">◆ Must – SSL socket negotiation is enabled.◆ None – Connection will not use SSL. Default.
SystemCertificateId	Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. Valid value is any alphanumeric string.
UsingRevealedPasswd	Indicates whether the password sent to the service is unobscured. Valid value is True or False. Default is False. Optional.

Business Process Example

The following example business process illustrates using the HTTP Client Begin Session service:

```
<process name="HTTPExample">
  <sequence>
    <operation name="Obscure Password">
      <!-- insert obscured password into process data -->
      <participant name="HTTPClientObscureParameter"/>
      <output message="outmsg">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientObscureResults" from="*"></assign>
      </input>
    </operation>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">hostb</assign>
        <assign to="RemotePort">26633</assign>
        <assign to="RemoteUserId">admin</assign>
        <!-- copy obscured password from process data to service -->
        <assign to="RemotePasswd" from="admin/text()"></assign>
        <assign to="SSL">Must</assign>
        <assign to="CipherStrength">Strong</assign>
        <assign to="CACertificateId">B2BHttp-Id</assign>
        <assign to="SystemCertificateId">httpstestclientcert1-Id</assign>
        -->
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

The following example shows how to use the revealObscured function when the user ID contains domain or special characters. First, create a name in the Obscure Service without any special characters and assign the appropriate password to it. In the following example, the name created in the Obscure Service is "htan" while the userid is sgp-htan\htan.

The parameter to the revealObscured() function is the node containing the obscured password. The function uses the node name as the key and the node value as the obscured password when unobscuring.

```
<operation name="Obscure HTTP client password">
  <participant name="HTTPClientObscureParameter"/>
  <output message="outmsg">
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="ObscureResult" from="*" />
  </input>
</operation>

<operation name=" HTTP Client Begin Session Service ">
```

```
<participant name="HTTPClientBeginSession" />
<output message="BeginSessionRequest">
    .....
    <assign to="RemoteUserId">sgp-htan\htan</assign>
    <assign to="UsingRevealedPasswd">>true</assign>
    <assign to="RemotePasswd" from="revealObscured(ObsecureResult/htan) " />
    .....
</output>
<input message="inmsg">
    <assign to=" HTTPClientBeginSessionServiceResults " from="*" />
</input>
</operation>
```

HTTP Client End Session Service

The following table provides an overview of the HTTP Client End Session service:

System name	HTTP Client End Session Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > HTTP Client
Description	This adapter is used to end an HTTP session with an external trading partner HTTP server. This service works through an instance of the HTTP Client adapter.
Business usage	A business user would use this service as the last functional activity in a business process that sends an HTTP request to a trading partner. This service can only be used if the HTTP Client Begin Session Service was used at the start of the business process.
Usage example	A Application business process is executed that translates a document that must be sent to a trading partner. After the translation, Application looks up information about how to transport data to the trading partner in the trading partner profile. The trading partner profile specifies HTTP as the transport protocol. Application then begins a session with the trading partner using the HTTP Client Adapter, sends the document, then ends the session using the HTTP Client End Session Service.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ HTTP Client adapter◆ HTTP Client Begin Session service◆ HTTP Client GET service◆ HTTP Client Method service◆ HTTP Client POST service
Application requirements	An HTTP server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ 0–Success◆ 1–Error
Restrictions	None

Persistence level	System Default
Testing considerations	To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the <i>HTTP Client adapter</i> . Debug information for this service can be found in the HTTP Client adapter and services log files.

Implementing the HTTP Client End Session Service

To implement the HTTP Client End Session service, complete the following tasks:

1. Create an HTTP Client End Session service configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client End Session service. For information, see *Configuring the HTTP Client End Session Service* on page 604.
3. Use the HTTP Client End Session service in a business process.

Configuring the HTTP Client End Session Service

To configure the HTTP Client End Session service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration.
SessionToken	Specifies the identifier of the session to be ended. Required.

Output from Service to Business Process

The following table contains the parameters passed from the HTTP Client End Session service to the business process:

Parameter	Description
SessionEndTime	Specifies the date and time that the session ended. Required.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client End Session service:

Parameter	Description
SessionToken	Specifies the ID of the session to be ended. Required.

Business Process Example

The following example business process illustrates using the HTTP Client End Session service:

```
<process name="default">
  <sequence>

    [[Insert Begin session operation here]]

    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken">SessionToken</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

HTTP Client GET Service

The following table provides an overview of the HTTP Client GET service:

System name	HTTP Client GET Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > HTTP Client
Description	The HTTP Client GET service sends HTTP GET requests to a trading partner's HTTP server via the perimeter server. This service works in conjunction with the HTTP Client Begin service and the HTTP Client End service, and through an instance of the HTTP Client adapter.
Business usage	A business user would use the HTTP Client GET service to retrieve documents from a trading partner's HTTP server.
Usage example	A Application business process is executed that must retrieve a specified file from the external trading partner. Application uses the HTTP Client GET Service, working through the HTTP Client Adapter, to retrieve the file from the trading partner's HTTP server. The data is passed to the business process.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms for Application
Related services	<ul style="list-style-type: none">◆ HTTP Client adapter◆ HTTP Client Begin Session service◆ HTTP Client End Session service◆ HTTP Client Method service◆ HTTP Client POST service
Application requirements	An HTTP server at the external trading partner location.
Initiates business processes?	No
Invocation	The HTTP Client GET service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 100s – Informational parameters and more status parameters will follow. ◆ 200s – Request was OK. ◆ 300s – Request failed and more status codes will follow. ◆ 400s – Client Request Failed. This is a client error. ◆ 500s – Server failed to handle a valid request. This is a server error.
Restrictions	None
Persistence level	Default
Testing considerations	<p>To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the <i>HTTP Client adapter</i>.</p> <p>Debug information for this service can be found in the HTTP Client adapter and services log files.</p>

Implementing the HTTP Client GET Service

To implement the HTTP Client GET service, complete the following tasks:

1. Create an HTTP Client GET service configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client GET service. For information, see *Configuring the HTTP Client GET Service* on page 607.
3. Use the HTTP Client GET service in a business process.

Configuring the HTTP Client GET Service

To configure the HTTP Client GET service, you must specify field settings in the Graphical Process Modeler (GPM):

Field	Description
Config	Name of the service configuration.
DocumentId	A document ID to use with this instance of the HTTP Client GET service. Optional.
RawResponse	<p>Whether to include the HTTP Server response headers in the response document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Both the HTTP headers and the entity body are copied to the body of the business process document. ◆ false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. <p>Default is false.</p>

Field	Description
ResponseTimeout	<p>Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.</p> <p>Note: The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration.</p>
SessionToken	<p>Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required.</p>
ShowResponseCode	<p>Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Metadata will be included. ◆ false – Metadata will not be included. Default is false.
URI	<p>Uniform Resource Indicator (URI) representing the HTTP server. Required.</p>
Cookie	<p>Identifies the cookie to use in cases where the previous GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Get Service uses the Cookie parameter and creates the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional.</p>

Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client GET service:

Field	Description
DocumentId	A document ID to use with this instance of the HTTP Client GET service. Optional.
RawResponse	Whether to include the HTTP Server response headers in the response document. Optional. Valid values: <ul style="list-style-type: none">◆ true – Both the HTTP headers and the entity body are copied to the body of the business process document.◆ false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. Default is false.
ResponseTimeout	Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid. Note: The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration.
SessionToken	Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required.
ShowResponseCode	Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values: <ul style="list-style-type: none">◆ true – Metadata will be included.◆ false – Metadata will not be included. Default is false.
URI	Uniform Resource Indicator (URI) representing the HTTP server. Required.
Cookie	Identifies the cookie to use in cases where the previous GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Get Service uses the Cookie parameter and creates the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional.

Business Process Examples

The following BPML example illustrates using commands supported by the HTTP Client GET service.

```
<process name="HTTPClientGETServiceExample">
  <sequence>

[[Insert begin session operation here]]

    <operation name="HTTP Client GET Service">
      <participant name="HTTPClientGETService"/>
      <output message="HTTPClientGetServiceTypeInputMessage">
        <assign to="DocumentId">DocumentID</assign>
      </output message>
    </operation>
  </sequence>
</process>
```

```

    <assign to="RawResponse">true</assign>
    <assign to="ResponseTimeout">60</assign>
    <assign to="SessionToken">SessionToken</assign>
    <assign to="ShowResponseCode">true</assign>
    <assign to="URI">URI</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

[[Insert end session operation here]]
</sequence>
</process>

```

The following business process provides an example of using the Cookie parameter:

```

<process name="HTTP_To_advancepcsr">
  <!-- Loop Invariant. -->
  <rule name="haveMoreCookie1">
    <condition> string(counterCookie) &lt;=
count(HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie/node())
</condition>
  </rule>

  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="CACertificateId">sgmillenia:13582d:10682043f1d:-73bd</assign>
        <assign to="CipherStrength">all</assign>
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="SSL">Must</assign>
        <assign to="RemoteHost">webtransport.advancepcsr.com</assign>
        <assign to="RemotePort">443</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>

    <!-- Use FSA to pick up the input file -->
    <operation name="Import Document Request">
      <participant name="TEST_FILE_SYSTEM_ADAPTER"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder" from="'/ais_local/share/sli'"></assign>
        <assign to="filter" from="'AdvancePCS_URI.txt'"></assign>
        <assign to="useSubFolders">false</assign>
        <assign to="bootstrap">false</assign>
        <assign to="deleteAfterCollect">false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="FileSystemOutputMessage">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

```

    </input>
</operation>

<!-- Set document content type/subtype -->
<operation name="SetContentType">
  <participant name="GetDocumentInfoService"/>
  <output message="xout">
    <assign to="." from="*"></assign>
    <assign to="DocumentContentType">application</assign>
    <assign to="DocumentContentSubType">x-www-form-urlencoded</assign>
  </output>
  <input message="xin">
    <assign to="docInfo" from="*"></assign>
  </input>
</operation>

<!-- POST to URI /template/login to login the webpage -->
<operation name="Http Client Post Service">
  <participant name="HTTPClientPost"/>
  <output message="HTTPClientPostServiceTypeInputMessage">
    <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="URI">/template/login</assign>
    <assign to="RawResponse">>true</assign>
    <assign to="ResponseTimeout">120</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientPostServiceResults" from="*" append="true"></assign>
  </input>
</operation>

<!-- concat the cookie from the POST Response -->
<assign to="counterCookie">1</assign>
<assign to="Cookie" from="''"/>
<choice>
<select>
<case ref="haveMoreCookie1" activity="AppendCookie1"/>
</select>
<sequence name="AppendCookie1">
<assign to="Cookie" from="concat(string(Cookie),
substring-before(//HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie[num
ber(//counterCookie)],';'),';')"/>
<assign to="counterCookie" from="number(counterCookie) + 1"/>
<choice>
<select>
<case ref="haveMoreCookie1" activity="Repeat1"/>
</select>
<repeat name="Repeat1" ref="AppendCookie1"/>
</choice>
</sequence>
</choice>

<!-- release Primary Document before GET -->
<operation>
<participant name="ReleaseService"/>

```

```

<output message="releaseRequest">
<assign to="TARGET" from="'PrimaryDocument'"/>
</output>
<input message="releaseResponse"/>
  </operation>

  <!-- GET URI / to reach the final page -->
  <operation name="Http Client Get Service">
    <participant name="HTTPClientGet"/>
    <output message="HTTPClientGetServiceTypeInputMessage">
      <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="URI"></assign>
      <assign to="RawResponse">true</assign>
      <assign to="ResponseTimeout">1200</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="HTTPClientGetServiceResults" from="*" append="true"></assign>
    </input>
  </operation>

  <operation name="HTTP Client End Session Service">
    <participant name="HTTPClientEndSession"/>
    <output message="HTTPClientEndSessionServiceTypeInputMessage">
      <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="HTTPClientEndSessionServiceResults" from="*"
append="true"></assign>
    </input>
  </operation>

  <onFault>
    <sequence name="End Session">
      <operation name="HTTP Client End Session Service">
        <participant name="HTTPClientEndSession"/>
        <output message="HTTPClientEndSessionServiceTypeInputMessage">
          <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
        </input>
      </operation>

    </sequence>
  </onFault>
</sequence>
</process>

```

The following business process is an example of using the URI parameter in the HTTP Client GET service. The URI is:

/s/ref=nb_ss_b/102-0129027-9554536?url=search-alias%3Dstripbooks&field-keywords=Computers&Go.x=6&G0.y=4

This URI has the following parameters:

Parameter	Description
url	search-alias%3Dstripbooks
field-keywords	Computers
Go.x	6
G0.y	4

```
<process name = "test_http_get">
  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">www.amazon.com</assign>
        <assign to="RemotePort">80</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="HTTP Client GET Service">
      <participant name="HTTPClientGet"/>
      <output message="HTTPClientGetServiceTypeInputMessage">
        <assign to="SessionToken" from="/ProcessData/SessionToken/text()"></assign>
        <assign
to="URI"/>/s/ref=nb_ss_b/102-0129027-9554536?url=search-alias%3Dstripbooks&field-
keywords=Computers&Go.x=6&Go.y=4</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken" from="/ProcessData/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
```

</process>

The resulting document in process data is the link to the Amazon page with the search requested.

Activity Types for the HTTP Client GET Service

The HTTP Client GET service reports the following activities to the Services Controller for service/GET service monitoring:

GET – Retrieves whatever information is identified by the URI in the request line.

HTTP Client Method Service

The following table provides an overview of the HTTP Client Method service:

System name	HTTP Client Method Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > HTTP Client
Description	The HTTP Client Method service sends HTTP requests to a trading partner's HTTP server via the perimeter server. The HTTP Client Method Service supports POST, GET, HEAD, or other valid HTTP request types that follow the normal HTTP request/response model. The actual HTTP methods available depends on the support provided by the remote/origin server. This service works in conjunction with the HTTP Client Begin service and the HTTP Client End service, and through an instance of the HTTP Client adapter.
Business usage	A business user would use the HTTP Client Method service to either retrieve or send data from Application to a trading partner when the HTTP protocol is required as the transport mechanism.
Usage example	A Application business process is executed that generates a document that must be sent to a trading partner using HTTP. Application initiates a session with the trading partner using the HTTP Client adapter and uses the HTTP Client Method Service, to place the document on the trading partner's HTTP server.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ HTTP Client adapter◆ HTTP Client Begin Session service◆ HTTP Client End Session service◆ HTTP Client GET service◆ HTTP Client POST service
Application requirements	An HTTP server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 100s – Informational parameters and more status parameters will follow. ◆ 200s – Request was OK. ◆ 300s – Request failed and more status codes will follow. ◆ 400s – Client Request Failed. This is a client error. ◆ 500s – Server failed to handle a valid request. This is a server error.
Restrictions	None
Persistence level	System Default
Testing considerations	<p>To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the <i>HTTP Client adapter</i>.</p> <p>Debug information for this service can be found in the HTTP Client adapter and services log files.</p>

Implementing the HTTP Client Method Service

To implement the HTTP Client Method service, complete the following tasks:

1. Create an HTTP Client Method service configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client Method service. For information, see *Configuring the HTTP Client Method Service* on page 616.
3. Use the HTTP Client Method service in a business process.

Configuring the HTTP Client Method Service

To configure the HTTP Client Method service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration.
DocumentId	A document ID to use with this instance of the HTTP Client Method service. Optional.
MethodType	Specifies the HTTP request type. Required. The HTTP Client Method Service supports POST, GET, HEAD, or other valid HTTP request types that follow the normal HTTP request/response model. The actual HTTP methods available depends on the support provided by the remote/origin server.

Field	Description
RawRequest	<p>Whether to include the HTTP header metadata in the primary document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – HTTP header metadata is included in the primary document. If set to true, the request document <i>must</i> contain headers inside its message body. ◆ false – HTTP header metadata is not included in the primary document. The headers are not available to the business process. <p>Default is false.</p> <p>Note: This parameter does not apply if MethodType is set to GET.</p>
RawResponse	<p>Whether to include the HTTP Server response headers in the response document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Both the HTTP headers and the entity body are copied to the body of the business process document. ◆ false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. <p>Default is false.</p>
ResponseTimeout	<p>Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.</p> <p>Note: The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration.</p>
SessionToken	<p>Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required.</p>
ShowResponseCode	<p>Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Metadata will be included. ◆ false – Metadata will not be included. Default is false.
URI	<p>Uniform Resource Indicator (URI) representing the HTTP server. Required.</p>
Cookie	<p>Identifies the cookie to use in cases where the previous POST or GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Method Service will use the Cookie parameter and create the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional.</p>

Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client Method service:

Field	Description
DocumentId	A document ID to use with this instance of the HTTP Client Method service. Optional.

Field	Description
MethodType	<p>Specifies the HTTP request type. Required. The HTTP Client Method Service supports POST, GET, HEAD, or other valid HTTP request types that follow the normal HTTP request/response model. The actual HTTP methods available depends on the support provided by the remote/origin server. Valid values include:</p> <ul style="list-style-type: none"> ◆ GET – Retrieves whatever information is identified by the URI in the request line. ◆ POST – Requests that the server accept the entity enclosed in the request as a new subordinate of the resource identified by the URI in the request line. ◆ HEAD – Retrieves the header of whatever information is identified by the URI in the request line.
RawRequest	<p>Whether to include the HTTP header metadata in the primary document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – HTTP header metadata is included in the primary document. If set to true, the request document <i>must</i> contain headers inside its message body. ◆ false – HTTP header metadata is not included in the primary document. The headers are not available to the business process. <p>Default is false.</p> <p>Note: This parameter does not apply if MethodType is set to GET.</p>
RawResponse	<p>Whether to include the HTTP Server response headers in the response document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Both the HTTP headers and the entity body are copied to the body of the business process document. ◆ false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. <p>Default is false.</p>
ResponseTimeout	<p>Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.</p> <p>Note: The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration.</p>
SessionToken	<p>Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required.</p>
ShowResponseCode	<p>Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Metadata will be included. ◆ false – Metadata will not be included. Default is false.
URI	<p>Uniform Resource Indicator (URI) representing the HTTP server. Required.</p>
Cookie	<p>Identifies the cookie to use in cases where the previous POST or GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Method Service will use the Cookie parameter and create the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional.</p>

Business Process Examples

The following example business process illustrates using commands supported by the HTTP Client Method service:

```
<process name="HTTPClientMethodServiceExample">
  <sequence>

    [[Insert Begin session operation here]]

    <operation name="HTTP Client Method Service">
      <participant name="HTTPClientMethodService"/>
      <output message="HTTPClientMethodServiceTypeInputMessage">
        <assign to="DocumentId">DocumentID</assign>
        <assign to="MethodType">HEAD</assign>
        <assign to="RawRequest">>true</assign>
        <assign to="ResponseTimeout">60</assign>
        <assign to="SessionToken">SessionToken</assign>
        <assign to="ShowResponseCode">>true</assign>
        <assign to="URI">URI</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    [[Insert End session operation here]]

  </sequence>
</process>
```

The following business process provides an example of using the Cookie parameter:

```
<process name="HTTP_To_advancepcsrx">
  <!-- Loop Invariant. -->
  <rule name="haveMoreCookie1">
    <condition> string(counterCookie) &lt;=
count(HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie/node())
</condition>
  </rule>

  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="CACertificateId">sgmillenia:13582d:10682043f1d:-73bd</assign>
        <assign to="CipherStrength">all</assign>
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="SSL">Must</assign>
        <assign to="RemoteHost">webtransport.advancepcsrx.com</assign>
        <assign to="RemotePort">443</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

```

    </input>
</operation>

<!-- Use FSA to pick up the input file -->
<operation name="Import Document Request">
  <participant name="TEST_FILE_SYSTEM_ADAPTER"/>
  <output message="FileSystemInputMessage">
    <assign to="Action">FS_COLLECT</assign>
    <assign to="collectionFolder" from="'/ais_local/share/sli'"></assign>
    <assign to="filter" from="'AdvancePCS_URI.txt'"></assign>
    <assign to="useSubFolders">false</assign>
    <assign to="bootstrap">false</assign>
    <assign to="deleteAfterCollect">false</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="FileSystemOutputMessage">
    <assign to="." from="*"></assign>
  </input>
</operation>

<!-- Set document content type/subtype -->
<operation name="SetContentType">
  <participant name="GetDocumentInfoService"/>
  <output message="xout">
    <assign to="." from="*"></assign>
    <assign to="DocumentContentType">application</assign>
    <assign to="DocumentContentSubType">x-www-form-urlencoded</assign>
  </output>
  <input message="xin">
    <assign to="docInfo" from="*"></assign>
  </input>
</operation>

<!-- POST to URI /template/login to login the webpage -->
<operation name="Http Client Post Service">
  <participant name="HTTPClientPost"/>
  <output message="HTTPClientPostServiceTypeInputMessage">
    <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="URI">/template/login</assign>
    <assign to="RawResponse">>true</assign>
    <assign to="ResponseTimeout">120</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientPostServiceResults" from="*" append="true"></assign>
  </input>
</operation>

<!-- concat the cookie from the POST Response -->
<assign to="counterCookie">1</assign>
<assign to="Cookie" from="''"/>
<choice>
<select>
<case ref="haveMoreCookie1" activity="AppendCookie1"/>
</select>

```

```

<sequence name="AppendCookie1">
  <assign to="Cookie" from="concat(string(Cookie),
  substring-before(//HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie[num
  ber(//counterCookie)],';'), '; ')" />
  <assign to="counterCookie" from="number(counterCookie) + 1" />
  <choice>
  <select>
  <case ref="haveMoreCookie1" activity="Repeat1" />
  </select>
  <repeat name="Repeat1" ref="AppendCookie1" />
  </choice>
</sequence>
  </choice>

  <!-- release Primary Document before GET -->
  <operation>
  <participant name="ReleaseService" />
  <output message="releaseRequest">
  <assign to="TARGET" from="'PrimaryDocument'" />
  </output>
  <input message="releaseResponse" />
  </operation>

  <!-- GET URI / to reach the final page -->
  <operation name="Http Client Get Service">
  <participant name="HTTPClientGet" />
  <output message="HTTPClientGetServiceTypeInputMessage">
  <assign to="SessionToken"
  from="HTTPClientBeginSessionServiceResults/SessionToken/text()" /></assign>
  <assign to="URI" /></assign>
  <assign to="RawResponse">true</assign>
  <assign to="ResponseTimeout">1200</assign>
  <assign to="." from="*" /></assign>
  </output>
  <input message="inmsg">
  <assign to="HTTPClientGetServiceResults" from="*" append="true" /></assign>
  </input>
  </operation>

  <operation name="HTTP Client End Session Service">
  <participant name="HTTPClientEndSession" />
  <output message="HTTPClientEndSessionServiceTypeInputMessage">
  <assign to="SessionToken"
  from="HTTPClientBeginSessionServiceResults/SessionToken/text()" /></assign>
  <assign to="." from="*" /></assign>
  </output>
  <input message="inmsg">
  <assign to="HTTPClientEndSessionServiceResults" from="*"
  append="true" /></assign>
  </input>
  </operation>

  <onFault>
  <sequence name="End Session">
  <operation name="HTTP Client End Session Service">
  <participant name="HTTPClientEndSession" />

```

```

        <output message="HTTPClientEndSessionServiceTypeInputMessage">
            <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
            <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
            <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
        </input>
    </operation>

</sequence>
</onFault>
</sequence>
</process>

```

The following business process provides an example of using the GetDocumentInfoService to set the content-type/subtype of the request document. It is important to put the output parameters of GetDocumentInfoService into a parent node, such as "docInfo".

The GetDocumentInfoService returns a node called DocumentId to the process data. This is the ID of the original document object without the content-type and sub-type set. If this node is directly under the root of Process Data, the HTTP Client Post service will attempt to retrieve the original document object using the document ID instead of the Primary Document. If DocumentId node is returned under "docinfo" node, the HTTP Client Post service does not use the document from DocumentId. Instead the service retrieves the correct document that was set with the correct content-type/subtype from the primary document.

```

<process name="HttpClient_SetContentType">
    <sequence>
        <!-- Set document content type/subtype -->
        <operation name="SetContentType">
            <participant name="GetDocumentInfoService"/>
            <output message="xout">
                <assign to="." from="*"></assign>
                <assign to="DocumentContentType">text</assign>
                <assign to="DocumentContentSubType">xml</assign>
            </output>
            <input message="xin">
                <assign to="docInfo" from="*"></assign>
            </input>
        </operation>

        <operation name="HTTP Client Begin Session Service">
            <participant name="HTTPClientBeginSession"/>
            <output message="HTTPClientBeginSessionServiceTypeInputMessage">
                <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
                <assign to="RemoteHost">10.235.18.103</assign>
                <assign to="RemotePort">37133</assign>
                <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
                <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
            </input>
        </operation>
    </sequence>
</process>

```

```

<operation name="Http Client Method Service">
  <participant name="HTTPClientMethod"/>
  <output message="HTTPClientMethodServiceTypeInputMessage">
    <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="MethodType">POST</assign>
    <assign to="URI">/hello</assign>
    <assign to="RawRequest">>false</assign>
    <assign to="RawResponse">>true</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientMethodServiceResults" from="*" append="true"></assign>
  </input>
</operation>

<operation name="HTTP Client End Session Service">
  <participant name="HTTPClientEndSession"/>
  <output message="HTTPClientEndSessionServiceTypeInputMessage">
    <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientEndSessionServiceResults" from="*"
append="true"></assign>
  </input>
</operation>

<onFault>
  <sequence name="End Session">
    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken"
from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
      </input>
    </operation>
  </sequence>
</onFault>
</sequence>
</process>

```

Activity Types for the HTTP Client Method Service

The HTTP Client Method service reports the following activities to the Services Controller for service/method service monitoring:

GET – Retrieves whatever information is identified by the URI in the request line.

POST – Requests that the server accept the entity enclosed in the request as a new subordinate of the resource identified by the URI in the request line.

HEAD – Retrieves the header of whatever information is identified by the URI in the request line.

The actual HTTP methods available depends on the support provided by the remote/origin server.

HTTP Client POST Service

The following table provides an overview of the HTTP Client POST service:

System name	HTTP Client POST Service
Graphical Process Modeler (GPM) categories	All Services, B2B Protocols > HTTP Client
Description	The HTTP Client POST Service sends HTTP POST requests to a trading partner's HTTP server via the perimeter server. This service works in conjunction with the HTTP Client Begin service and the HTTP Client End service, and through an instance of the HTTP Client adapter.
Business usage	A business user would use the HTTP Client POST Service to place documents on a trading partner's HTTP server.
Usage example	A business process is executed that generates a document to be transferred to a trading partner using the HTTP protocol. The application initiates a session with the trading partner using the HTTP Client Adapter and uses the HTTP Client POST Service to place the data on the trading partner's HTTP server.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms for Application
Related services	<ul style="list-style-type: none">◆ HTTP Client adapter◆ HTTP Client Begin Session service◆ HTTP Client End Session service◆ HTTP Client GET service◆ HTTP Client Method service
Application requirements	An HTTP server must exist at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ 100s – Informational parameters and more status parameters will follow. ◆ 200s – Request was OK. ◆ 300s – Request failed and more status codes will follow. ◆ 400s – Client Request Failed. This is a client error. ◆ 500s – Server failed to handle a valid request. This is a server error.
Restrictions	None
Persistence level	Default
Testing considerations	<p>To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the <i>HTTP Client adapter</i>.</p> <p>Debug information for this service can be found in the HTTP Client adapter and services log files.</p>

Implementing the HTTP Client POST Service

To implement the HTTP Client POST service, complete the following tasks:

1. Create an HTTP Client POST service configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client POST service. For information, see *Configuring the HTTP Client POST Service* on page 626.
3. Use the HTTP Client POST service in a business process.

Configuring the HTTP Client POST Service

To configure the HTTP Client POST service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration.
DocumentId	A document ID to use with this instance of the HTTP Client POST service. Optional.
RawRequest	<p>Whether to include the HTTP header metadata in the primary document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – HTTP header metadata is included in the primary document. If set to true, the request document <i>must</i> contain headers inside its message body. ◆ false – HTTP header metadata is not included in the primary document. The headers are not available to the business process. <p>Default is false.</p> <p>Note: This parameter does not apply if MethodType is set to GET.</p>

Field	Description
RawResponse	<p>Whether to include the HTTP Server response headers in the response document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Both the HTTP headers and the entity body are copied to the body of the business process document. ◆ false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. <p>Default is false.</p>
ResponseTimeout	<p>Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.</p> <p>Note: The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration.</p>
SessionToken	<p>Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required.</p>
ShowResponseCode	<p>Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:</p> <ul style="list-style-type: none"> ◆ true – Metadata will be included. ◆ false – Metadata will not be included. Default is false.
URI	<p>Uniform Resource Indicator (URI) representing the HTTP server. Required.</p>
Cookie	<p>Identifies the cookie to use in cases where the previous POST needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Post Service uses the Cookie parameter and creates the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional.</p>

Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client POST service:

Field	Description
DocumentId	A document ID to use with this instance of the HTTP Client POST service. Optional.
RawRequest	Whether to include the HTTP header metadata in the primary document. Optional. Valid values: <ul style="list-style-type: none">◆ true – HTTP header metadata is included in the primary document. If set to true, the request document <i>must</i> contain headers inside its message body.◆ false – HTTP header metadata is not included in the primary document. The headers are not available to the business process. Default is false. Note: This parameter does not apply if MethodType is set to GET.
RawResponse	Whether to include the HTTP Server response headers in the response document. Optional. Valid values: <ul style="list-style-type: none">◆ true – Both the HTTP headers and the entity body are copied to the body of the business process document.◆ false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. Default is false.
ResponseTimeout	Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid. Note: The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration.
SessionToken	Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required.
ShowResponseCode	Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values: <ul style="list-style-type: none">◆ true – Metadata will be included.◆ false – Metadata will not be included. Default is false.
URI	Uniform Resource Indicator (URI) representing the HTTP server. Required.
Cookie	Identifies the cookie to use in cases where the previous POST needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Post Service uses the Cookie parameter and creates the Cookie header in the next request.

Business Process Examples

The following example business process illustrates using commands supported by the HTTP Client POST service:

```

<process name="HTTPClientPOSTServiceExample">
  <sequence>

    [[Insert Begin session operation here]]

    <operation name="HTTP Client POST Service">
      <participant name="HTTPClientPOSTService"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="DocumentId">DocumentID</assign>
        <assign to="RawRequest">true</assign>
        <assign to="RawResponse">true</assign>
        <assign to="ResponseTimeout">60</assign>
        <assign to="SessionToken">SessionToken</assign>
        <assign to="ShowResponseCode">true</assign>
        <assign to="URI">URI</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    [[Insert End session operation here]]

  </sequence>
</process>

```

The following business process provides an example of using the Cookie parameter:

```

<process name="HTTP_To_advancepcsrx">
  <!-- Loop Invariant. -->
  <rule name="haveMoreCookie1">
    <condition> string(counterCookie) &lt;=
      count(HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie/node())
    </condition>
  </rule>

  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="CACertificateId">sgmillenia:13582d:10682043f1d:-73bd</assign>
        <assign to="CipherStrength">all</assign>
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="SSL">Must</assign>
        <assign to="RemoteHost">webtransport.advancepcsrx.com</assign>
        <assign to="RemotePort">443</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>

    <!-- Use FSA to pick up the input file -->
    <operation name="Import Document Request">
      <participant name="TEST_FILE_SYSTEM_ADAPTER"/>

```

```

<output message="FileSystemInputMessage">
  <assign to="Action">FS_COLLECT</assign>
  <assign to="collectionFolder" from="'/ais_local/share/sli'"></assign>
  <assign to="filter" from="'AdvancePCS_URI.txt'"></assign>
  <assign to="useSubFolders">>false</assign>
  <assign to="bootstrap">>false</assign>
  <assign to="deleteAfterCollect">>false</assign>
  <assign to="." from="*"></assign>
</output>
<input message="FileSystemOutputMessage">
  <assign to="." from="*"></assign>
</input>
</operation>

<!-- Set document content type/subtype -->
<operation name="SetContentType">
  <participant name="GetDocumentInfoService"/>
  <output message="xout">
    <assign to="." from="*"></assign>
    <assign to="DocumentContentType">application</assign>
    <assign to="DocumentContentSubType">x-www-form-urlencoded</assign>
  </output>
  <input message="xin">
    <assign to="docInfo" from="*"></assign>
  </input>
</operation>

<!-- POST to URI /template/login to login the webpage -->
<operation name="Http Client Post Service">
  <participant name="HTTPClientPost"/>
  <output message="HTTPClientPostServiceTypeInputMessage">
    <assign to="SessionToken"
      from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="URI">/template/login</assign>
    <assign to="RawResponse">>true</assign>
    <assign to="ResponseTimeout">120</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientPostServiceResults" from="*" append="true"></assign>
  </input>
</operation>

<!-- concat the cookie from the POST Response -->
<assign to="counterCookie">1</assign>
<assign to="Cookie" from="'"></assign>
<choice>
<select>
<case ref="haveMoreCookie1" activity="AppendCookie1"/>
</select>
<sequence name="AppendCookie1">
<assign to="Cookie" from="concat(string(Cookie),
substring-before(//HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie[number(//counterCookie)],';'), '; ')">
<assign to="counterCookie" from="number(counterCookie) + 1"/>
<choice>

```

```

<select>
<case ref="haveMoreCookie1" activity="Repeat1"/>
</select>
<repeat name="Repeat1" ref="AppendCookie1"/>
</choice>
</sequence>
  </choice>
<!-- release Primary Document before GET -->
  <operation>
<participant name="ReleaseService"/>
<output message="releaseRequest">
<assign to="TARGET" from="'PrimaryDocument'"/>
</output>
<input message="releaseResponse"/>
  </operation>

  <!-- GET URI / to reach the final page -->
  <operation name="Http Client Get Service">
    <participant name="HTTPClientGet"/>
    <output message="HTTPClientGetServiceTypeInputMessage">
      <assign to="SessionToken"
        from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="URI"></assign>
      <assign to="RawResponse">true</assign>
      <assign to="ResponseTimeout">1200</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="HTTPClientGetServiceResults" from="*" append="true"></assign>
    </input>
  </operation>

  <operation name="HTTP Client End Session Service">
    <participant name="HTTPClientEndSession"/>
    <output message="HTTPClientEndSessionServiceTypeInputMessage">
      <assign to="SessionToken"
        from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="HTTPClientEndSessionServiceResults" from="*"
        append="true"></assign>
    </input>
  </operation>

  <onFault>
    <sequence name="End Session">
      <operation name="HTTP Client End Session Service">
        <participant name="HTTPClientEndSession"/>
        <output message="HTTPClientEndSessionServiceTypeInputMessage">
          <assign to="SessionToken"
            from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
        </input>
      </operation>
    </sequence>
  </onFault>

```

```

        </input>
    </operation>

    </sequence>
</onFault>
</sequence>
</process>

```

The following business process provides an example of using the `GetDocumentInfoService` to set the content-type/subtype of the request document. It is important to put the output parameters of `GetDocumentInfoService` into a parent node, such as "docInfo".

The `GetDocumentInfoService` returns a node called `DocumentId` to the process data. This is the ID of the original document object without the content-type and sub-type set. If this node is directly under the root of Process Data, the HTTP Client Post service will attempt to retrieve the original document object using the document ID instead of the Primary Document. If `DocumentId` node is returned under "docinfo" node, the HTTP Client Post service does not use the document from `DocumentId`. Instead the service retrieves the correct document that was set with the correct content-type/subtype from the primary document.

```

<process name="HttpClient_SetContentType">
  <sequence>
    <!-- Set document content type/subtype -->
    <operation name="SetContentType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="." from="*"></assign>
        <assign to="DocumentContentType">text</assign>
        <assign to="DocumentContentSubType">xml</assign>
      </output>
      <input message="xin">
        <assign to="docInfo" from="*"></assign>
      </input>
    </operation>

    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">10.235.18.103</assign>
        <assign to="RemotePort">37133</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>

    <operation name="Http Client Post Service">
      <participant name="HTTPClientPost"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="SessionToken"
          from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="URI">/hello</assign>
        <assign to="RawRequest">>false</assign>
        <assign to="RawResponse">>true</assign>
      </output>
    </operation>
  </sequence>
</process>

```



```

    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientPostServiceResults" from="*" append="true"></assign>
  </input>
</operation>

<operation name="HTTP Client End Session Service">
  <participant name="HTTPClientEndSession"/>
  <output message="HTTPClientEndSessionServiceTypeInputMessage">
    <assign to="SessionToken"
      from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="HTTPClientEndSessionServiceResults" from="*"
      append="true"></assign>
  </input>
</operation>

<onFault>
  <sequence name="End Session">
    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken"
          from="HTTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
      </input>
    </operation>
  </sequence>
</onFault>
</sequence>
</process>

```

The following business process provides an example of specifying parameters in URI of the HTTP Client POST service. The URI is:

```

/gettraf/portal_gettraf/processGentranData?UID=f87db70048484b0fe6348eaebbf622
81&status=0&errorMsg=Erro

```

The URI in this example has three parameters:

Parameter	Value
ei	utf-8
fr	slv8-msgr
p	http%20POST%20examples

The business process is the following:

```

<process name = "test_http_post">
  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">search.yahoo.com</assign>
        <assign to="RemotePort">80</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="HTTP Client POST Service">
      <participant name="HTTPClientPost"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="RawRequest">true</assign>
        <assign to="SessionToken" from="/ProcessData/SessionToken/text()"></assign>
        <assign to="ShowResponseCode">true</assign>
        <assign
to="URI">/search?ei=utf-8&fr=slv8-msgr&p=http%20POST%20examples</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken" from="/ProcessData/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Activity Types for the HTTP Client POST Service

The HTTP Client POST service reports the following activities to the Services Controller for service/POST service monitoring:

POST – Requests that the server accept the entity enclosed in the request as a new subordinate of the resource identified by the URI in the request line.

HTTP Respond Service

The following table provides an overview of the HTTP Respond service:

System name	HTTP Respond Service
Graphical Process Modeler (GPM) category	None
Description	<p>This adapter sends responses to trading partners who have submitted an HTTP request. The request could have been submitted through the B2B HTTP Server adapter or through the new, Perimeter server-based HTTP Server adapter.</p> <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. For more information on the retirement process, see <i>Retiring and Removed Services and Adapters</i>.</p>
Business usage	A business user would use this adapter to return a response to a trading partner over the HTTP transport protocol, when the trading partner initiates the HTTP connection.
Usage example	A trading partner submits a document to a URL in Application. The URL is configured to initiate a specific business process. The business process completes activities and returns a response to the trading partner to confirm that the request was processed. This response is returned using this service.
Preconfigured?	Yes
Requires third party files?	When used with the HTTP Server adapter, Jetty's two jar files must be available; Jetty is an open source, embeddable webserver and servlet engine (http://jetty.mortbay.org).
Platform availability	All supported Application platforms
Related services	HTTP Server adapter
Application requirements	Unless the Perimeter server local-mode is used, the external Perimeter server must be running.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	<ul style="list-style-type: none">◆ Success – Normal execution with the return parameters specified below.◆ Error – For any error cases, namely missing parameter and adapter not found.
Restrictions	None
Persistence level	None
Testing considerations	Debug information related to this adapter can be found in <code>http.log</code> . The level of logging can be controlled through the Application UI or the <code>log.properties</code> file.

How the HTTP Respond Service Works

The following example illustrates how the HTTP Respond service works:

1. A trading partner sends data to your company using HTTP.
2. Your HTTP Server adapter receives the data, initiates a business process, and writes the transport-instance-id and transport-session-id for the HTTP connection into process data.
3. The business process runs and after specified tasks are completed, invokes the HTTP Respond service.
4. The business process passes the transport-instance-id and transport-session-id from the original HTTP connection to the service.
5. The service uses these IDs to find the original HTTP connection, which is still open, and uses it to send a reply to your trading partner.

Implementing the HTTP Respond Service

A configuration of the HTTP Respond service (HttpRespond) is provided with Application. Because it has no configuration parameters, it should not be necessary to create any new configurations of the HTTP Respond service. The configuration provided should suffice for all uses of this service, including using the service in a business process.

Output from Business Process to Service

The following table describes the output from the business process to the HTTP Respond service:

Parameter	Description
transport-instance-id	Identifies the specific instance of the adapter on which the request was received. This might be a B2B HTTP Server adapter or a Perimeter server-based HTTP Server adapter. Valid value is a non-empty string generated by an HTTP server adapter. Required. Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. For more information on the retirement process, see <i>Retiring and Removed Services and Adapters</i> .
transport-session-id	Transport Session ID. Identifies the specific inbound connection on the HTTP Server adapter on which the request was received and to which the response must be returned. In either case, the adapter is identified by the transport instance ID. Valid value is a non-empty string that should not be fabricated. Each value is created by an HTTP Server adapter instance for an inbound HTTP session. Required.
doc-has-headers	Raw mode selection for the response. Valid values are True and False. If True, when the service runs, the primary document is expected to have the HTTP headers for the response; that is, the primary document has a MIME, RFC822 or HTTP structure with headers and body. If False, the primary document does not contain any headers. Required.

Parameter	Description
status-code	The HTTP status code for the response message to the HTTP client. Each range has a specific meaning. For example, codes between 200 and 299 are success codes and those between 400 and 599 specify different kinds of failures. Valid values are between 100 and 599. Default is 200. Optional.

Business Process Example

The following example business process returns an HTTP response with the contents of the primary document being the HTTP body (non-raw mode):

```
<process name="NonRawHttpResponse">
  <sequence>
    <operation>
      <participant name="HttpResponse"/>
      <output message="noopout">
        <assign to="doc-has-headers">false</assign>
        <assign to="." from="*" />
      </output>
      <input message="noopin">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

HTTP Reverse Proxy Adapter

The HTTP Reverse Proxy adapter provides a high level of data protection between external connections using HTTP and your company's Application server. Use this adapter to receive documents from a trading partner who uses the HTTP protocol.

The HTTP Reverse Proxy adapter can only be used with the Sterling Secure Proxy 2.0 (or later) product. See the Sterling Secure Proxy documentation for information.

HTTP Server Adapter

The following table provides an overview of the HTTP Server adapter.

System name	HTTP Server Adapter
Graphical Process Modeler (GPM) category	None. Cannot be used as part of a business process.
Description	<p>Processes HTTP requests from trading partners using a Perimeter server.</p> <p>The HTTP Server adapter in the application includes a bundled WebDAV server that is licensed independently. For information about using WebDAV in the application, see the <i>WebDAV Server</i> documentation.</p> <p>The HTTP Server adapter replaces the B2B HTTP Server adapter, which is being retired. The HTTP Server adapter and its related services provide all of the functionality of that adapter, plus these benefits:</p> <ul style="list-style-type: none">◆ Uses Perimeter services◆ Uses the same Jetty HTTP server engine as the application's ASI console◆ Able to run both WARs and BPML applications◆ Runs \inside the application's JVM for access to all application resources
Business usage	Use this adapter to send documents to and receive documents from a trading partner using HTTP.
Usage example	A trading partner submits a document to a URL in the application. The URL is configured to initiate a specific business process. The business process completes activities and returns a success response to the trading partner to confirm that the request was processed. If the URL is configured for a web application, the identified web application is started and delegated to. At that point the flow is determined by the web application.
Preconfigured?	Three instances of this adapter, named HTTP Server adapter, MBI HTTP Server adapter, and RN HTTP Server adapter, are installed with the application. By default they use a local-mode Perimeter server.
Requires third party files?	Must obtain two .jar files from Jetty; Jetty is an open source, embeddable Web server and servlet engine (http://jetty.mortbay.org). Two .jar files for Jasper (an open source JSP compilation library) and the .jar file for Ant (an open source build and compilation library) must be available for the handling of web applications that contain JSPs.
Platform availability	All supported application platforms
Related services	This adapter must be used in conjunction with the HTTP Respond service (which is the only way to return an HTTP response to a request waiting at a particular adapter instance).
Application requirements	When this adapter is configured with a non-local-mode Perimeter server, the Perimeter server must be installed and running. This Perimeter server is typically installed in a DMZ environment, separated from the application by a firewall.
Initiates business processes?	This adapter can find the name of a business process that is configured for a particular URL, initiate that business process and wait for the response.

Invocation	Is not invoked by a business process. To return a response, use the HTTP Respond service.
Business process context considerations	When a business process is initiated as a result of an HTTP request, the initial context process data contains the transport-instance-id and transport-session-id, information necessary for the HTTP Respond service to return the HTTP response. Process data also contains any query parameters in the URL.
Returned status values	None
Restrictions	WAR file deployment functionality is not available for WebSphere.
Persistence level	None
Testing considerations	Debug information related to this adapter can be found in http.log.

How the HTTP Server Adapter Works

The HTTP Server adapter receives data from a trading partner using HTTP. This adapter differs from the existing HTTP adapter (B2B HTTP Server adapter) in how it works with secure installations that use a DMZ. The older B2B HTTP Server adapter requires an HTTP servlet to be installed in your company DMZ; instead of using a servlet, this adapter communicates with a Perimeter server that has been installed in the DMZ. Using the HTTP Server adapter, you can initiate a business process or a web application. If you are initiating a business process and want to send a reply or status to your trading partner about this data, you must have the business process invoke the HTTP Respond service, which sends the response to your trading partner.

Note: The HTTP Server adapter replaces the B2B HTTP Server adapter, which has entered the retirement process. For more information on the retirement process, see *Retiring and Removed Services and Adapters*.

Example

Your trading partner sends EDI purchase orders to your company, using HTTP and sending the data to a URI that you have specified.

You set up a configuration of the HTTP Server adapter that is associated with that URI. As part of creating this configuration, you specify whether the URI will be associated with a business process or a web application, then select the specific business process or web application.

Your trading partner sends data to your URI, using an HTTP request. When the request is received and passed to the HTTP Server adapter, it invokes the business process or web application that you specified when creating the adapter configuration. The adapter collects the transport-instance-id and transport-session-id from the initial request, and places the information into process data.

If the adapter initiates a business process, the HTTP request connection is put into a wait state while the business process completes. Once complete, the HTTP Respond service is called and uses the transport-instance-id and transport-session-id that were stored in process data to send a reply on the same connection that the request came in on.

If the adapter initiates a web application, the web application determines what to return and when.

Implementing the HTTP Server Adapter

To implement the HTTP Server adapter, complete the following tasks:

1. Create an HTTP Server adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Server adapter. For information, see *Configuring the HTTP Server Adapter*.

Note: If configuring the HTTP Server adapter for WebDAV, see the specific configuration instructions in the *WebDAV Server* documentation.

Configuring the HTTP Server Adapter

To configure the HTTP Server adapter, you must specify field settings in the application as described in the following table.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Leave this set to: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time. Note: Do not use the HTTP Server adapter in groups.
HTTP Listen Port	The port number on which the Perimeter server process listens for connections from external trading partner HTTP clients. If a local-mode Perimeter server is chosen, this listen port is bound on the local computer. Valid values are 1 through 65536. On many operating systems, only the root user can bind on ports 1 through 1024. Required.
Perimeter Server Name	List of available Perimeter servers, including local-mode Perimeter servers. Required. Default is local-mode Perimeter server.
Document Storage	Where to store the body of the request document. Valid values are: <ul style="list-style-type: none">◆ System Default◆ Database◆ File System Default is System Default. Required. Note: For more information about document storage types, see <i>Managing Services and Adapters</i> .
User Authentication Required	Whether to enable HTTP basic authentication. Valid values are: <ul style="list-style-type: none">◆ Yes – A connection must pass HTTP basic authentication to be serviced.◆ No – HTTP basic authentication is not to be used. Default is Yes. Required.

Field	Description
Use SSL	Whether SSL Server authentication must be enabled or not. Valid values are: <ul style="list-style-type: none"> ◆ Must – SSL is enabled ◆ None – SSL is disabled Default is None. Required.
System Certificate	Select a system certificate from the list. This is the private key that the SSL server will use. Required if Use SSL is Must.
Cipher Strength	Specifies the strength of the algorithms (cipher suites) used to encrypt data. Valid values are: <ul style="list-style-type: none"> ◆ STRONG – Required if Use SSL is Must ◆ ALL – All cipher strengths are supported ◆ WEAK – Often required for international trade, because government regulations prohibit STRONG encryption from being exported Default is STRONG. Required if SSL is checked.
CA Certificate	Move one or more CA Certificates to the use column. These are the digital security certificates that the SSL server will use to authenticate the client. Optional.
URI	Uniform Resource Indicator (URI) representing incoming requests. Add one or more URIs to represent incoming requests and the business process or web application (represented as a WAR file) associated with each. Required.
Launch BP or WAR	Whether the URI launches a business process or WAR file. Default is Business Process. Required.
Enter WAR File Path	Specifies WAR file to be launched by URI. Valid value is any accessible path. Required if WAR File is selected for Launch BP or WAR File field.
Business Process	Specifies business process to be launched by URI. Select from the list of available business processes. Required if BP is selected for Launch BP or WAR File field.
Send Raw Messages	Whether the raw message is presented to the business process. The term raw denotes that the primary document associated with the business process contains HTTP headers. Valid values are: <ul style="list-style-type: none"> ◆ Yes – Both the HTTP headers and the entity body are copied to the body of the business process document before the business process is started. This setting is required for EDIINT AS2, RosettaNet, and ebXML. ◆ No – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. Default is No. Required if BP is selected for Launch BP or WAR File field.
Run BP in sync mode	Whether to invoke Web services in synchronous mode. Valid values are: <ul style="list-style-type: none"> ◆ Yes – HTTP Server Adapter bootstraps the BP in synchronous mode. HTTP Server Adapter executes the BP in the same thread. ◆ No – HTTP Server Adapter bootstraps the BP asynchronous mode. Default is No. Required if BP is to be run in synchronous mode.

Output from Adapter to Business Process

The following table describes the output from the HTTP Server adapter to the business process.

Field Name	Description
http-request-uri	Target URI as specified by the Trading Partner. Required for applications that need it. For example, SOAP.
transport-instance-id	Identifies the specific instance of the HTTP Server adapter on which the request was received. Valid value is a non-empty string that should not be fabricated. Each value is created for an HTTP Server adapter instance. Required.
transport-session-id	Transport Session ID. Identifies the specific inbound connection on the HTTP Server adapter (identified by the transport instance id) on which the request was received and to which the response must be returned. Valid value is a non-empty string that should not be fabricated. Each value is created by an HTTP Server adapter instance for an inbound HTTP session. Required.
b2b-protocol	Identifies the protocol type, with a value of http.
SyncModeBP	Indicates whether the BP has been bootstrapped in synchronous mode or not. Valid values are true and false.

Initial Process Data XML Document Example

The following example shows how the initial process data XML document looks:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <PrimaryDocument SCIObjectID="server1:b1aebf:fa40ae79ca:-7209" />
  <b2b-protocol>http</b2b-protocol>
  <transport-instance-id>TestHTTPServerAdapter-insecure_HttpAdapter_node1</transport-
instance-id>
  <transport-session-id>Thu Jan 22 22:04:16 EST 2004:5</transport-session-id>
  <http-request-uri>/reflect</http-request-uri>
</ProcessData>
```

Activity Types for This Service

This adapter reports the following activities to the Services Controller for Service/Adapter Monitoring:

Get—retrieves whatever information is identified by the http-request-uri.

Post—requests that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the URI in the Request-Line.

Enhancing HTTP Server Adapter Performance

To improve performance, the HTTP Server adapter enables you to specify a range of threads for handling events. The range, which is specified in the http.properties file, includes a Min Thread value and Max

Thread value. If the Max Thread value is reached, any additional connection requests fail. The http.properties file is located in the properties folder under your application installation directory.

Human Interaction Document Loader Service

The Human Interaction Document Loader service retrieves a document previously saved in the Application database by the Human Interaction Event service. After the document is retrieved, the Human Interaction Document Loader service loads the document into the business process as the primary document. The following table provides an overview of the Human Interaction Document Loader service:

System name	HumanInteractionDocumentLoader
Graphical Process Modeler (GPM) categories	All Services, Web Extensions
Description	<p>Uses a valid MinedDataId of a previously saved document to retrieve the document from the Application database.</p> <p>Note: The MinedDataId is created when a document is added to the database using the Human Interaction Event service, and corresponds to the value in the MINED_DATA_ID column of the WEBX_MINED_DATA table.</p>
Business usage	<p>A business process uses the Human Interaction Event service to save a document requiring approval in the database and then sends out an e-mail notification to the approver. The e-mail notification includes the URL with the Web template, along with the MinedDataId, for retrieving and displaying the document. The Web template includes a call to a business process that uses the Human Interaction Document Loader service to retrieve the document and the B2B HTTP Server adapter for displaying the document in the Web browser for approval or rejection.</p> <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See <i>Retiring and Removed Services and Adapters</i>.</p>
Usage example	<p>The individual responsible for approving a purchase order does the following:</p> <ol style="list-style-type: none">1 Receives an e-mail notification with a URL link for displaying the purchase order.2 Launches a Web browser and specifies the URL link.3 Approves or rejects the purchase order displayed in the Web browser.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms

Related services	Designed to work with: <ul style="list-style-type: none"> ◆ Human Interaction Query service ◆ B2B Lookup System service ◆ B2B HTTP Server adapter ◆ HTTP Server adapter ◆ Human Interaction XForms service <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See <i>Retiring and Removed Services and Adapters</i>.</p>
Application requirements	None
Initiates business processes?	None
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ WorkFlowContext.SUCCESS: Success, with no errors. ◆ WorkFlowContext.ERROR NoMinedDataIDException: Caught NoMinedDataIDException in DocumentLoaderService.processData()
Restrictions	None
Persistence level	None
Testing considerations	None

How the Human Interaction Document Loader Service Works

Use the Human Interaction Document Loader service to retrieve documents from the Application database. The following sections describe a business scenario and a sample solution using the Human Interaction Document Loader service.

Business Scenario

Your company receives purchase orders in XML format from a trading partner. All purchase orders exceeding \$1,000.00 need to be approved. You want to view a purchase order requiring approval in the Web browser.

Business Solution Example

The approach used to solve this business scenario includes creating a business process that includes:

- A Human Interaction Document Loader service to retrieve the purchase order requiring approval
- A B2B HTTP Server adapter or HTTP Server adapter to display the purchase order in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

Business Solution Example Dependencies

The following dependencies must be met to run the example in this section and display the purchase order in a Web browser:

Run a business process that accepts a purchase order as input and uses the Human Interaction Event service to store the purchase order in the Application database and mark it as requiring approval. The Human Interaction Event service must also be configured to invoke a business process that constructs and sends an e-mail notification containing the URL for displaying the purchase order to the approver. The URL must contain the name of the Web template, along with the document ID for the purchase order requiring approval. See the business scenario and business solution example for the B2B Lookup System service for an example business process.

Deploy a Web template in Application that invokes the business process in this example and displays the purchase order in the Web browser.

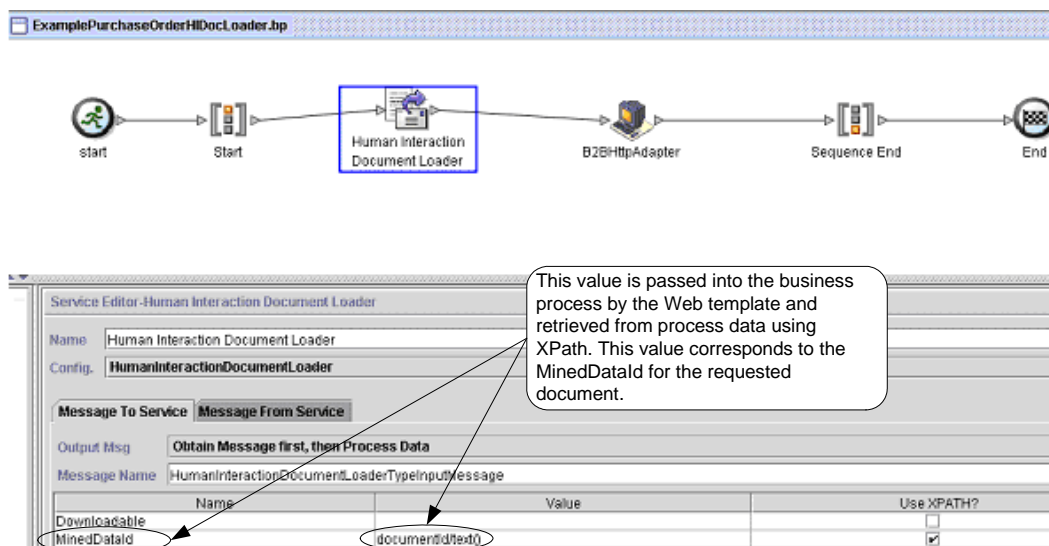
Create a configuration of the HTTP Respond service that specifies the URI used to invoke the business process created in this example from the Web browser.

This business solution example focuses only on the Human Interaction Document Loader service functionality.

GPM Example

The following example shows a solution to the business scenario using the GPM.

Note: This example assumes that the Web template used to display the purchase order is deployed in Application.



The Human Interaction Event service (not shown) generated an e-mail notification to the approver after the purchase order was added to the database. The approver uses this URL to display the purchase order requiring approval in the Web browser. For example:

```
http://siHostName:siport/websuite/xfm/
ExamplePurchaseOrderFromHTTPRequest.xfm?documentId=7736bd:fb78c70638:-77df
```

where:

ExamplePurchaseOrderFromHTTPRequest.xfm is the name of the Web template used to display the purchase order. This Web template is designed to invoke the business process that retrieves the purchase order from the Application database.

documentId is set to the MinedDataId (7736bd:fb78c70638:-77df) of the requested document in the Application database.

The business process run by the Web template uses the Human Interaction Document Loader service to retrieve from the database the purchase order that corresponds to the documentId, and uses the B2B HTTP Server adapter to display the purchase order in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

After the Human Interaction Document Loader service completes, the purchase order is returned to the business process as the primary document, along with a document named HumanInteractionEvents, which contains the reference data from the Human Interaction Event service for the purchase order. The returned Human Interaction Events document for this example looks like the following:

```
<list xmlns="" >
<listItem objectId="7736bd:fb78c70638:-77df">
<referenceId>200</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>admin</systemAccount>
<templateName>ExamplePurchaseOrderFromHTTPRequest</templateName>
<date>2004-03-23T18:24:05.000Z</date>
<waiting>NO</waiting>
<wfcId>E1700-000284:7736bd:fb78c70638:-77e1</wfcId>
<workflowId>57029</workflowId>
<fromAccount>admin</fromAccount>
<fromMinedDataId>7736bd:fb78c70638:-77df</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<status>Unread</status>
<state>Awaiting Approval</state>
<timeout/>
<modifiedDate>2004-03-23T18:24:05.000Z</modifiedDate>
<marked/>
</listItem>
</list>
```

The returned primary document (purchase order) for this example looks like the following:

```
<Order Id="200">
  <Order_Item Id="400">
```



```

<Order_Id>200</Order_Id>
<Name>DSL</Name>
<Quantity>1</Quantity>
<Price> $1050.00</Price>
</Order_Item>
<Order_Item Id="401">
<Order_Id>200</Order_Id>
<Name>Cable</Name>
<Quantity>1</Quantity>
<Price> $1500.00</Price>
</Order_Item>
</Order>

```

The primary document is then sent to the B2B HTTP Server adapter for display in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*. For example, you might have a Web template that displays the purchase order in the following HTML format:

PURCHASE ORDER REQUESTS			
<u>Order Number</u>	<u>Buyer</u>	<u>Request Date</u>	<u>State</u>
PO1234	TradingPartner1	03/20/2004	Awaiting Approval
PO4567	TradingPartner2	03/21/2004	Awaiting Approval

The approver can choose to approve or reject the purchase order.

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML.

```

<process name="ExamplePurchaseOrderHIDocLoader">
<sequence name="Start">
<operation name="Human Interaction Document Loader">
<participant name="HumanInteractionDocumentLoader"/>
<output message="HumanInteractionDocumentLoaderTypeInputMessage">
  <assign to="MinedDataId" from="documentId/text()"/>
  <assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
<operation name="HTTP Respond Service">
  <participant name="HttpRespond"/>

```

```

        <output message="HttpRespondServiceInputMessage">
            <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
            <assign to="." from="*"></assign>
        </input>
    </operation>
</sequence>
</process>

```

Implementing the Human Interaction Document Loader Service

To implement the Human Interaction Document Loader service, complete the following tasks:

1. Activate your license for the Human Interaction Document Loader service. See *Installing Application*.
2. Create a Human Interaction Document Loader service configuration. See *Managing Services and Adapters*.
3. Use the Human Interaction Document Loader service in a business process. For information, see *Human Interaction Document Loader Service Business Process Usage*.

Human Interaction Document Loader Service Business Process Usage

The following screen shows a graphical view of the GPM parameters for the Human Interaction Document Loader service. There are no fields to be configured on the Message From Service tab.

Name	Value	Use XPath?
Downloadable	Yes	<input type="checkbox"/>
MinedDataId	documentId/text()	<input checked="" type="checkbox"/>

The following example shows the corresponding BPML parameters for the Human Interaction Document Loader service GPM parameters.

```

<process name="ExamplePurchaseOrderHIDocLoader">
<sequence name="Start">
<operation name="Human Interaction Document Loader">
<participant name="HumanInteractionDocumentLoader"/>
<output message="HumanInteractionDocumentLoaderTypeInputMessage">
    <assign to="Downloadable">YES</assign>
    <assign to="MinedDataId" from="documentId/text()"/>
    <assign to="." from="*" />
</output>
<input message="inmsg">
    <assign to="." from="*" />
</input>

```

```

</operation>
</sequence>
</process>

```

The following table describes the fields used to configure the Human Interaction Document Loader service in the GPM. The values specified for these parameters are the input to the Human Interaction Document Loader service from the business process.

Field	Description
Config	Name of the service configuration.
Downloadable	<p>Whether a file download box displays when trying to download files. This parameter applies to Web Extension applications that provide a means to download, save, or display the content of a file saved in a workflow.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Always displays the file download box each time you download a file to your client computer, regardless of file type. This gives you the option of saving the file to disk or displaying the file contents in the browser regardless of the file type. ◆ No – Determines whether or not to display a file download box based on the file content. For example, if the downloaded file is in XML format, the browser displays the raw XML in the browser rather than prompting you to save the file. If the downloaded file is text format, the browser prompts you to either display or save the file to disk.
MinedDataId	<p>Unique identifier that is associated with a document within a business process and corresponds to the value in the MINED_DATA_ID column of the WEBX_MINED_DATA table in the Application database. This value is populated when a document is added to the database using the Human Interaction Event service. Valid value is a MinedDataId that exists in the WEBX_MINED_DATA table.</p> <p>Note: If you want to reuse the business process and service, you should not hardcode this value. Instead, use an XPath expression to set this parameter.</p>

Output from Service to Business Process

The following table describes the output from the Human Interaction Document Loader service to the business process:

Returned Output	Description
HumanInteractionEvents/@SCIObjctID document	The name of the document, along with the document SCI Object ID, returned by the Human Interaction Document Loader service. This document contains the reference data from the Human Interaction Event service for the specific document retrieved from the Application database.

Human Interaction Event Service

The Human Interaction Event service flags business processes that require human action and creates references to the business process data to await human action.

The Human Interaction Event service:

Is the service that starts human interaction at one or more steps in the business process.

Handles the updating and deletion of mined data, which is data pulled from the business process and written to the Application internal database for future use through demining. Demining is the action of retrieving data from the internal database for display in a Web browser.

Is used at any point in a business process where human interaction is needed.

The following table provides an overview of the Human Interaction Event service:

System name	HumanInteractionEvent
Graphical Process Modeler (GPM) categories	All Services, Web Extensions
Description	<p>The Human Interaction Event service:</p> <ul style="list-style-type: none">◆ Adds documents requiring Human Interaction to the WEBX_MINED_DATA table in the Application database. In addition to adding the document, reference data about the document and the current state of the business process used to invoke the Human Interaction Event service is added to the database.◆ Updates documents, along with the document reference data in the database.◆ Deletes documents from the database, along with the document reference data. <p>This service can also suspend a business process until certain actions take place (such as approval of a document) and then resume processing.</p>
Business usage	<p>A business process uses the Human Interaction Event service to save documents requiring approval in the database and then send out an e-mail notification to the approver. Another business process might include the following services and adapters to work with the documents saved in the database:</p> <ul style="list-style-type: none">◆ Human Interaction Query service to retrieve all documents requiring approval from the database◆ Human Interaction XForms service and B2B HTTP Server adapter to display the results from the Human Interaction Query service to the Web browser for user selection <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See <i>Retiring and Removed Services and Adapters</i>.</p> <ul style="list-style-type: none">◆ Human Interaction Document Loader service to retrieve the selected document from the database and pass the document to the Human Interaction XForms service for displaying in the Web browser for approval

Usage example	An organization may want to monitor all purchase orders, requiring approval for any PO request that exceeds \$1000.00. The business process associated with the submission of a PO could be configured to send the PO request to the employee's manager, suspending the business process until the approval is complete. After the PO is approved or rejected, the business process completes the steps to either submit the PO or send a rejection back to the employee.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Designed to work with: <ul style="list-style-type: none"> ◆ Human Interaction Query service ◆ Human Interaction Document Loader service ◆ Human Interaction XForms service
Application requirements	None
Initiates business processes?	The user can specify the WebSuiteEmailNotif business process on the Human Interaction Event service NotificationBPName parameter to invoke this business process for generating an e-mail notification. The WebSuiteEmailNotif business process comes predefined with Application. The user can also create a customized business process that sends e-mail notifications in a specific format and invokes the business process using the NotificationBPName parameter.
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ WorkflowContext.SUCCESS: Success, with no errors. ◆ WorkflowContext.ERROR IllegalArgumentException: Caught IllegalArgumentException in EventService.processData() ◆ WorkflowContext.ERROR IllegalStateException: Caught IllegalStateException in EventService.processData() ◆ WorkflowContext.ERROR SQLException: Caught SQLException in EventService.processData()
Restrictions	None
Testing considerations	None

How the Human Interaction Event Service Works

Use the Human Interaction Event service to save business process data requiring human interaction in the Application database. The following sections describe a business scenario and a sample solution for using the Human Interaction Event service.

Business Scenario

Your company receives a purchase order in XML format from a trading partner. The purchase order needs to be approved prior to further processing.

The approach used to solve this business scenario includes the creation of a business process that does the following:

- Accepts the purchase order as input.

- Includes a Human Interaction Event service that saves a reference to the purchase order, the reference data about the purchase order, and the current state of the business process data in the Application database and marks the business process as requiring human interaction. The reference data for this example includes the following information:

- ◆ The document type (purchase order)
- ◆ The Application user account of the person sending the purchase order
- ◆ The Application user account of the person who is to receive the purchase order (approver)
- ◆ The state of the purchase order (Awaiting Approval)
- ◆ The status of the purchase order (Unread)
- ◆ The Web template used to display the purchase order in the Web browser

The Human Interaction Event service is also configured to invoke the WebSuiteEmailNotif business process for sending an e-mail notification to the approver. The e-mail notification includes the URL used to display the purchase order requiring approval in the Web browser.

Note: You must have a valid, checked-in Web template to display an item requiring human interaction in the Web browser.

This business solution example focuses only on the Human Interaction Event service functionality.

GPM Example

The following example shows a solution to the business scenario using the GPM. The purchase order is input to the business process and becomes the primary document.

ExamplePurchaseOrderHIEventWithMail.bp

Service Editor: Human Interaction Event

Name: Human Interaction Event
 Config: HumanInteractionEvent

Message To Service: Message From Service

Output Msg: Obtain Message first, then Process Data
 Message Name: HumanInteractionEventInputMessage

Name	Value	Use XPath?
ArchiveFlag		<input type="checkbox"/>
ArchiveHours		<input type="checkbox"/>
DocumentType	Purchase Order	<input type="checkbox"/>
FromAccount	Sender UserAccount	<input type="checkbox"/>
GroupName		<input type="checkbox"/>
IdentityName		<input type="checkbox"/>
MergeDataOnResume		<input type="checkbox"/>
MergeDataOnUpdate		<input type="checkbox"/>
NotificationBPName	WebSuiteEmailNotif	<input type="checkbox"/>
Operation	ADD	<input type="checkbox"/>

The primary document and the parameter settings are input to this service.

Business process used to construct and send an e-mail notification that includes the URL for displaying the purchase orders to the approver.

(Screen 1 of 2)

The **SystemAccount** is the approver's user account in Sterling Integrator. The e-mail notification is sent to the e-mail address associated with this account.

The **TemplateName** is the valid, checked-in Web template used to display the purchase order.

ReferenceId	number(/ProcessData/poNumber/@id)	<input checked="" type="checkbox"/>
State		<input type="checkbox"/>
Status	Awaiting Approval	<input type="checkbox"/>
StorageArea	Inbox	<input type="checkbox"/>
SuspendAsWaiting	No	<input type="checkbox"/>
SystemAccount	ReceiverSterlingIntegratorUserAccount	<input type="checkbox"/>
TemplateName	ExamplePurchaseOrderFromHttpRequest	<input type="checkbox"/>
Timeout		<input type="checkbox"/>

(Screen 2 of 2)

The Assign statement is used to assign the purchase order number from the primary document to process data. The Human Interaction Event service uses the purchase order number in process data to set its ReferenceId parameter.

After the Human Interaction Event service completes, a row is added to the Application WEBX_MINED_DATA database table containing the reference data about the purchase order and a

reference to the actual purchase order document. In addition, an e-mail notification is sent to the approver that includes the URL for displaying the actual purchase order. For example:

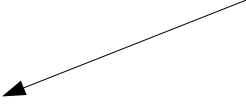
STERLING Integrator - Web Suite Email Notification

You have received a new Purchase Order in your Web Suite Inbox.

To: ReceiverUserAccount
 From: SenderUserAccount
 RE: Purchase Order - 200
 Location: Inbox
 Timestamp: 2004-12-17T18:49:55.000Z
 Expires (minutes):

<http://00.000.0.000:00000/websuite/xfm/ExamplePurchaseOrderFromHTTPRequest.xfm?documentId=1446e4:fb582af649:-5dcc>

The WebSuiteEmailNotif business process, which was set with the Notification BPName parameter, is used to construct this URL.



Note: The Web template specified in the URL should be designed to invoke a business process that uses the Human Interaction Document Loader service to load the purchase order (identified by its documentId in the URL) from the Application database prior to displaying the Web page.

The approver specifies the URL in the Web browser to display the purchase order requiring approval. The approver could alternatively log into the Application Web Suite application to view (in their inbox) a list of documents requiring approval.

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML:

```
<process name="ExamplePurchaseOrderHIEventWithMail">
<sequence name="Start HI Event Service">
<assign name="Assign" to="poNumber" from="DocToDOM(PrimaryDocument)/@Id"
append="true"/>
<operation name="Human Interaction Event">
<participant name="HumanInteractionEvent"/>
<output message="HumanInteractionEventTypeInputMessage">
  <assign to="DocumentType">Purchase Order</assign>
  <assign to="FromAccount">SenderSterlingIntegratorUserAccount</assign>
  <assign to="NotificationBPName">WebSuiteEmailNotif</assign>
  <assign to="Operation">ADD</assign>
  <assign to="ReferenceId" from="number(/ProcessData/poNumber/@Id)"/>
  <assign to="Status">Awaiting Approval</assign>
  <assign to="StorageArea">Inbox</assign>
  <assign to="SuspendAsWaiting">NO</assign>
  <assign to="SystemAccount">ReceiverSterlingIntegratorUserAccount</assign>
  <assign to="TemplateName">ExamplePurchaseOrderFromHTTPRequest</assign>
  <assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
</sequence>
```


</process>

See *Usage Examples* for additional examples using the Human Interaction Event service.

Implementing the Human Interaction Event Service

To implement the Human Interaction Event service, complete the following tasks:

1. Activate your license for the Human Interaction Event service. See *Installing Application*.
2. Create a Human Interaction Event service configuration. See *Managing Services and Adapters*.
3. Use the Human Interaction Event service in a business process.

Human Interaction Event Service Business Process Usage

The following screen shows the graphical view of the GPM parameters for the Human Interaction Event service. There are no fields to be configured on the Message From Service tab.

Name	Value	Use XPath?
ArchiveFlag	1	<input type="checkbox"/>
ArchiveHours	24	<input type="checkbox"/>
DocumentType	Purchase Order	<input type="checkbox"/>
FromAccount	SendingTradingPartnerSystemAccountID	<input type="checkbox"/>
GroupName	SendingTradingPartnerSecurityPermissionGroup	<input type="checkbox"/>
IdentityName	SendingTradingPartnerName	<input type="checkbox"/>
MergeDataOnResume	No	<input type="checkbox"/>
MergeDataOnUpdate	No	<input type="checkbox"/>
NotificationBPName	UserDefinedBPForSendingEmailNotifications	<input type="checkbox"/>
Operation	ADD	<input type="checkbox"/>
ParentAccount	ManagerOfTheDocumentSender	<input type="checkbox"/>
ReferenceId	number(ProcessData/poNumber/@id)	<input checked="" type="checkbox"/>
State	PendingApproval	<input type="checkbox"/>
Status	Unread	<input type="checkbox"/>
StorageArea	Inbox	<input type="checkbox"/>
SuspendAsWalling	No	<input type="checkbox"/>
SystemAccount	ReceiverCompanyUserAccount	<input type="checkbox"/>
TemplateName	ExamplePurchaseOrders	<input type="checkbox"/>

The following example shows the corresponding BPML parameters for the Human Interaction Event service GPM parameters.

```
<process name="ExampleHIEEventGMPParams">
  <sequence name="Start HI Event Service">
    <operation name="Human Interaction Event">
      <participant name="HumanInteractionEvent" />
      <output message="HumanInteractionEventTypeInputMessage">
        <assign to="ArchiveFlag">1</assign>
        <assign to="ArchiveHours">24</assign>
        <assign to="DocumentType">Purchase Order</assign>
        <assign to="FromAccount">SendingTradingPartnerSystemAccountID</assign>
        <assign to="GroupName">SendingTradingPartnerSecurityPermissionGroup</assign>
        <assign to="IdentityName">SendingTradingPartnerName</assign>
      </output message>
    </operation>
  </sequence>
</process>
```

```

    <assign to="MergeDataOnResume">NO</assign>
    <assign to="MergeDataOnUpdate">NO</assign>
    <assign
to="NotificationBPName">UserDefinedBPForSendingEmailNotifications</assign>
    <assign to="Operation">ADD</assign>
    <assign to="ParentAccount">ManagerOfTheDocumentSender</assign>
    <assign to="ReferenceId" from="number(/ProcessData/poNumber/@Id)"/>
    <assign to="State">PendingApproval</assign>
    <assign to="Status">Unread</assign>
    <assign to="StorageArea">Inbox</assign>
    <assign to="SuspendAsWaiting">NO</assign>
    <assign to="SystemAccount">ReceiverCompanyUserAccount</assign>
    <assign to="TemplateName">ExamplePurchaseOrders</assign>
    <assign to="Timeout"/>
    <assign to="." from="*" />
</output>
<input message="inmsg">
    <assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

The following table describes the fields used to configure the Human Interaction Event service in the GPM:

Field	Description
Config	Name of the service configuration.
ArchiveHours	The number of hours the mined data record should remain in the Web Extension table before it is archived or purged. Valid value is a positive integer. Optional. If blank, the time span from the business process is used, if it has been specified. If the time span from the business process has not been specified, the system default is used.
ArchiveFlag	The archive method that should be used. Optional. Valid values: <ul style="list-style-type: none"> ◆ Archive – Archives the data in the Web Extension table. ◆ Purge – Deletes the data from the Web Extension table. If left blank, the archive method from the business process is used.
DocumentType	Type of data that requires human interaction. Valid value is any alphanumeric string. Examples include Purchase Order, Invoice, and Remittance Advice. Note: The value specified for this parameter is the name of the document that appears under Document Type in the user's inbox when using the Application Web Suite to display documents.
FromAccount	Trading partner system user account ID (as defined in Application) of the trading partner who sent the document. This account is also associated with the trading partner's trading profile.
GroupName	Name of the security permission group to which the trading partner who sent the document belongs.

Field	Description
IdentityName	Name of the trading partner who sent the document. This name is associated with the FromAccount information listed in this service. This parameter allows for searching records pertaining to a specific company.
MergeDataOnResume	Merges any changes made to the Web template with the existing instance data when the business process is taken out of the waiting state. Valid values: <ul style="list-style-type: none"> ◆ YES – Causes the documents and instance data to be merged when the business process is resumed. ◆ NO – Overlays the business process being resumed (taken out of a waiting state) with the changed documents and instance data. Default.
MergeDataOnUpdate	Merges any changes made to the Web template with the existing instance data. Valid values: <ul style="list-style-type: none"> ◆ YES – Default. ◆ NO – Update only the status or state of an item and not the document contents. The parameter affects processing only when the Operation parameter is set to UPDATE.
NotificationBPName	Business process (for example, the WebSuiteEmailNotif business process) runs by the Human Interaction Event service with the purpose of generating an e-mail notification to the individual who receives the document. The WebSuiteEmailNotif generates an e-mail notification based on the e-mail address associated with the Application user account (SystemAccount) set up for the trading partner receiving the document and the user account (FromAccount) set up for the trading partner who sent the document. The WebSuiteEmailNotif business process comes predefined with Application. Alternatively, you can create and specify a customized business process that sends e-mail notifications in a specific format. If you choose this method, you should model the WebSuiteEmailNotif business process. Valid values are any checked-in business process from the list that is designed to generate e-mail notifications. The default value is Not Applicable. Note: If you are using a business process that looks up the e-mail addresses from the user accounts specified with the SystemAccount and FromAccount parameters and if the Application user account does not contain a valid e-mail address, the associated SMTP Send adapter fails and the e-mail notification is not sent.
Operation	The requested action for the document. Required. Valid values: <ul style="list-style-type: none"> ◆ ADD – Add a new document along with its reference data. ◆ DELETE – Delete an existing document along with its reference data. ◆ UPDATE – Update an existing document along with its reference data. Note: You must specify the MinedDataId parameter as an input parameter to the Human Interaction Event service for the DELETE and UPDATE operations. Use the Advanced Editor or BPML to specify the MinedDataId parameter. See <i>Updating or Deleting a Document in the Application Database</i> for more information about the MinedDataId and examples for specifying the parameter.
ParentAccount	Account name for the manager of the From Account—the document sender. This user account name is associated with the FromAccount information in this service.

Field	Description
ReferenceID	Document identifier. Valid value is any alphanumeric string, such as a purchase order number or an invoice number. Note: The value specified for this parameter is displayed as the Identifier in the user's inbox when using the Application Web Suite to display documents.
State	Document process stage. Valid value is any alphanumeric string, such as Approved, Pending, or Rejected. If no state is entered, the default is Active.
Status	Status of the document. Valid value is any alphanumeric string, such as Read or Unread. Note: The value specified for this parameter is displayed as the Status in the user's inbox when using the Application Web Suite to display documents.
StorageArea	Virtual location to store the data waiting for the person to take action on it. Valid value is any alphanumeric string, such as Inbox, Outbox, and Drafts. Note: The value specified for this parameter corresponds to a folder name used to display documents from this storage area in the Application Web Suite.
SuspendAsWaiting	Setting for specifying whether to place the business process into a waiting status until human interaction is complete. Required if Operation field is set to ADD or UPDATE. Valid values: <ul style="list-style-type: none"> ◆ Yes – Business process is placed in a wait state. The business process continues waiting until an action is taken on the document (such as updating by another instance of the Human Interaction Event service) or until the value specified with the Timeout parameter expires. See <i>Placing a Business Process in a Waiting State for Human Interaction</i> for an example. ◆ No – Business process continues processing and the document becomes available for viewing.
SystemAccount	Trading partner user account ID (as defined in Application) of the person who must interact with the business process. Note: The document is stored in this user's storage area (such as the inbox) in the Application Web Suite.
TemplateName	Name of the Web template to use for viewing the data. Select the Web template name from the list of deployed Web templates. Required if you are using the Application Web Suites to view documents. Note: If you are using a custom Web application for viewing data in the Application database, you can either specify the Web template with the Human Interaction Event service or with the Human Interaction XForms service.
Timeout	Amount of time, in minutes, before the waiting status expires, causing the business process to fail with an advanced status of Human Interaction Timeout Event. Timeout is recognized only if the SuspendAsWaiting field is set to YES. Note: The Timeout parameter also is dependent on the BPExpirator. The BPExpirator schedule looks for expired business processes based on the time set in the scheduler and resumes the business process with an advanced status. The default schedule time is 15 minutes.

Usage Examples

The following sections contain additional examples for using the Human Interaction Event service.

Placing a Business Process in a Waiting State for Human Interaction

The example in this section describes a scenario that uses the Human Interaction Event service to save a reference to the business process data in the database and place the business process in a wait state until the purchase order is approved. The Human Interaction Event service is also configured to invoke the WebSuiteEmailNotif to send an e-mail notification to the approver. After the purchase order is approved, the business process resumes and continues with the next step. In this scenario, the next step is to invoke a subprocess for further processing of the purchase order.

The following example illustrates this scenario using the GPM.

The screenshot displays the Service Editor for a Human Interaction Event. The top part shows a process flow diagram with the following steps: start, Start HI Event Service, Assign, Human Interaction Event (highlighted with a blue box), Invoke Sub-Process (circled in red), Sequence End, and End. Below the flow is the configuration panel for the 'Human Interaction Event' service.

Service Editor - Human Interaction Event

Name: Human Interaction Event
 Config: HumanInteractionEvent

Message To Service: Message From Service

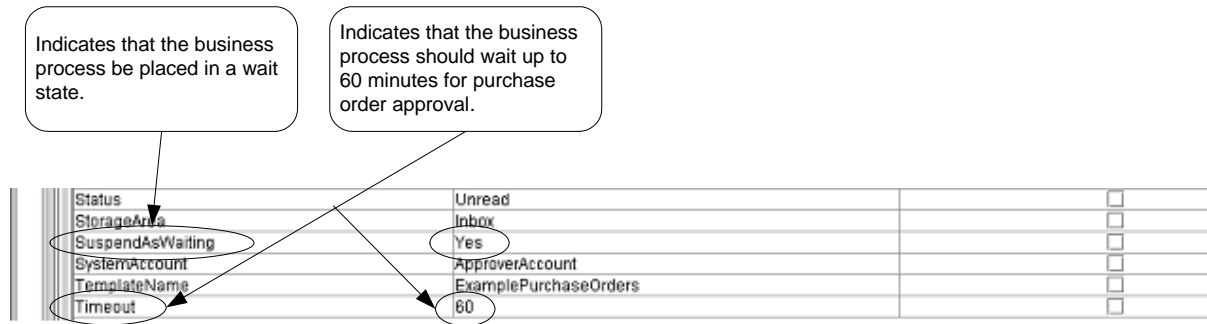
Output Msg: Obtain Message first, then Process Data
 Message Name: HumanInteractionEventInputMessage

Name	Value	Use XPATH?
DocumentType	Purchase Order	<input type="checkbox"/>
FromAccount	SendingTradingPartner	<input type="checkbox"/>
GroupName		<input type="checkbox"/>
IdentityName		<input type="checkbox"/>
MergeDataOnResume	No	<input type="checkbox"/>
MergeDataOnUpdate	No	<input type="checkbox"/>
NotificationBPName	WebSuiteEmailNotif	<input type="checkbox"/>
Operation	ADD	<input type="checkbox"/>
ParentAccount		<input type="checkbox"/>
Referenceld	number(ProcessData/poNumber/@Id)	<input checked="" type="checkbox"/>
State	PendingApproval	<input type="checkbox"/>

Annotations:

- A callout box points to the 'Invoke Sub-Process' step in the flow diagram, stating: "Indicates that a subprocess is invoked after the purchase order is approved and the business process resumes processing."
- Two callout boxes point to the 'NotificationBPName' and 'WebSuiteEmailNotif' values in the configuration table, stating: "Business process used to construct and send an e-mail notification to the approver (SystemAccount)".

Screen 1 of 2



(Screen 2 of 2)

The Assign statement is used to assign the purchase order number from the primary document to process data. The Human Interaction Event service uses the purchase order number in process data to set its ReferenceId parameter. After the document is added to the database, you can use the following services to complete the approval process:

- Human Interaction Query service to retrieve a list of purchase order references awaiting approval

- Human Interaction Document Loader service to retrieve the actual purchase order associated with the reference returned from the Human Interaction Query service

- Human Interaction XForms service to display the purchase order in the browser for necessary updates and approval

- Human Interaction Event service to update the database with the changes made to the purchase order and to change the status of the purchase order to Approved. After updating the purchase order, the originating business process resumes processing and starts the subprocess. See *Updating or Deleting a Document in the Application Database* for more information.

The following example illustrates the business process using BPML:

```

<process name="ExamplePurchaseOrderHIEventWaiting">
  <sequence name="Start HI Event Service">
    <assign to="poNumber" from="DocToDOM(PrimaryDocument)/@Id" append="true" />
    <operation name="Human Interaction Event">
      <participant name="HumanInteractionEvent" />
      <output message="HumanInteractionEventTypeInputMessage">
        <assign to="DocumentType">Purchase Order</assign>
        <assign to="FromAccount">SendingTradingPartner</assign>
        <assign to="MergeDataOnResume">NO</assign>
        <assign to="MergeDataOnUpdate">NO</assign>
        <assign to="NotificationBPName">WebSuiteEmailNotif</assign>
        <assign to="Operation">ADD</assign>
        <assign to="ReferenceId" from="number(/ProcessData/poNumber/@Id)"/>
        <assign to="State">PendingApproval</assign>
        <assign to="Status">Unread</assign>
        <assign to="StorageArea">Inbox</assign>
        <assign to="SuspendAsWaiting">YES</assign>
        <assign to="SystemAccount">ApproverAccount</assign>
        <assign to="TemplateName">ExamplePurchaseOrders</assign>
      </output message>
    </operation>
  </sequence>
</process>

```

```

    <assign to="Timeout">60</assign>
    <assign to="." from="*" />
</output>
<input message="inmsg">
    <assign to="." from="*" />
</input>
</operation>
<operation name="Invoke Sub-Process">
<participant name="InvokeSubProcessService" />
<output message="InvokeSubProcessServiceTypeInputMessage">
    <assign to="INVOKE_MODE">ASYNC</assign>
    <assign to="WFD_NAME">ExampleHIEEventSubProcess</assign>
    <assign to="." from="*" />
</output>
<input message="inmsg">
    <assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

For more information, see:

Human Interaction Query Service for a business scenario describing how to use this service to retrieve the purchase order reference data from the Application database.

Human Interaction Document Loader Service for a business scenario describing how to use this service to retrieve the actual purchase order from the Application database.

Human Interaction XForms Service for a business scenario describing how to use this service to display a list of documents.

Updating or Deleting a Document in the Application Database

You must specify the MinedDataId parameter as an input parameter to the Human Interaction Event service when updating or deleting a document in the database. The MinedDataId parameter corresponds to the value in the MINED_DATA_ID column of the WEBX_MINED_DATA table. This value is populated when a document is added to the database using the Human Interaction Event service.

Use the Human Interaction Query service to retrieve the reference data for documents. The reference data includes the MinedDataId for each document. The MinedDataId corresponds to the listItem objectId attribute returned by the Human Interaction Query service.

You can specify these fields using the Advanced Editor option in the GPM Service Editor or by using BPML.

The following example illustrates specifying the MinedDataId parameter in the GPM. This example assumes that the MinedDataId value was loaded into process data (not shown).

The screenshot shows the Service Editor interface for a Human Interaction Event. The main window displays the event configuration, including the name "Human Interaction Event" and the configuration "HumanInteractionEvent". The "Advanced Editor" dialog box is open, showing a table with the following data:

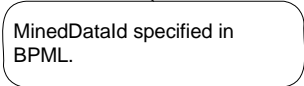
Name	Value	Use XPath?
MinedDataId	string(ProcessData['MinedDataId@object'])	<input checked="" type="checkbox"/>

Annotations in the image include:

- A callout box pointing to the "Advanced Editor" dialog box with the text: "In the Advanced Editor, specify the MinedDataId parameter and its value."
- A callout box pointing to the "Advanced" button in the bottom right corner with the text: "Click **Advanced** to display the Advanced Editor."

The following example illustrates the same business process using BPML. This example assumes that the MinedDataId value was loaded into process data (not shown).

```
<process name="ExamplePurchaseOrderHIQueryAndHIXformSTUB">
<sequence name="Start">
<operation name="Human Interaction Event">
<participant name="HumanInteractionEvent"/>
<output message="HumanInteractionEventTypeInputMessage">
  <assign to="DocumentType">Purchase Order</assign>
  <assign to="FromAccount">Joe Supplier</assign>
  <assign to="Operation">UPDATE</assign>
  <assign to="State">Approved</assign>
  <assign to="Status">Read</assign>
  <assign to="StorageArea">Inbox</assign>
  <assign to="SuspendAsWaiting">NO</assign>
  <assign to="SystemAccount">admin</assign>
  <assign to="TemplateName">ExamplePurchaseOrders</assign>
  <assign to="MinedDataId" from="string(/ProcessData/list/listItem/@objectId)"/>
  <assign to="." from="*" />
</output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
</sequence>
</process>
```



Note: If the business process used to add the document to the database is in a wait state, an update or delete causes the business process to resume processing.

Human Interaction Query Service

The Human Interaction Query service searches Application for business process data requiring human interaction and returns the query results to the business process. The query results include the reference data for documents that were previously saved in the Application database by the Human Interaction Event service. The query results also include the information about the state of the business process that was used to save the documents in the database (for example, whether or not the business process is in a waiting state).

Typically, after the Human Interaction Query service has completed, the Human Interaction XForms service identifies the Web template used to display the results returned from the service. The Human Interaction XForms service then works in conjunction with the B2B HTTP Server adapter to display the results in a Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

The following table provides an overview of the Human Interaction Query service:

System name	HumanInteractionQuery
Graphical Process Modeler (GPM) categories	All Services, Web Extensions
Description	Searches the Application WEBX_MINED_DATA table for records that match the query parameters specified for this service. The results from the query are saved in XML format and returned to the business process as a document.
Business usage	<p>A business process uses the Human Interaction Event service to save documents requiring approval in the database and then sends out an e-mail notification to the approver. The e-mail notification includes the URL with the Web template for displaying the documents. When the approver specifies the URL in the browser, the Web template invokes a business process that uses the Human Interaction Query service to retrieve all documents requiring approval from the database and uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the results in the Web browser for user selection.</p> <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See <i>Retiring and Removed Services and Adapters</i>.</p>

Usage example	<p>A manager logs into the Application Web extensions and wants to view all items in her inbox requiring approval. The process might include the following steps:</p> <ol style="list-style-type: none"> 1 The manager specifies a URL in the Web browser that includes the name of the Web template used to display the list of items. 2 The Web template invokes a business process that uses the Human Interaction Query service to retrieve items from the manager's inbox prior to being displayed. 3 The Web template is displayed with the list of items requiring approval by the manager. 4 The manager selects and approves or rejects an item from the list.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>Designed to work with:</p> <ul style="list-style-type: none"> ◆ B2B Lookup System service ◆ Human Interaction Document Loader service ◆ Human Interaction XForms service
Application requirements	None
Initiates business processes?	None
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ <code>WorkflowContext.SUCCESS</code>: Success, with no errors. There still might not be a result set, but no errors encountered by service. For example, the service may not find rows in the <code>WEBX_MINED_DATA</code> table meeting the query criteria and therefore no results are returned to the business process. ◆ <code>WorkflowContext.ERROR IllegalArgumentException</code>: Caught <code>IllegalArgumentException</code> in <code>QueryService.processData()</code> ◆ <code>WorkflowContext.ERROR IllegalStateException</code>: Caught <code>IllegalStateException</code> in <code>QueryService.processData()</code> ◆ <code>WorkflowContext.ERROR TransformerException</code>: Caught <code>TransformerException</code> in <code>QueryService.processData()</code>
Restrictions	None
Persistence level	None
Testing considerations	None

Requirements

Using this service requires:

- Enabling the Human Interaction Event service configuration to store documents requiring Human Interaction in the Application database

- Enabling the Human Interaction XForms service configuration to retrieve items requiring Human Interaction from the Application database

- Creating a valid, deployed Web template for displaying data in a Web browser

How the Human Interaction Query Service Works

Use the Human Interaction Query service to search the Application database for business process data that requires human interaction. The query results returned from this service include the reference data for documents that were previously saved in the Application database by the Human Interaction Event service.

Business Scenario

Your company receives purchase orders in XML format from a trading partner. All purchase orders exceeding \$1,000.00 need to be approved. You want to view a list of all purchase orders requiring approval.

Business Solution Example

The approach used to solve this business scenario includes creating a business process that includes:

- A Human Interaction Query service to retrieve the list of purchase orders requiring approval

- A Human Interaction XForms service to receive the list of purchase orders from the Human Interaction Query service and identify the Web template used to display the list.

- A B2B HTTP Server adapter for displaying the list in a Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

Business Solution Example Dependencies

The following dependencies must be met to run the example in this section and display the results in a Web browser:

- Run a business process that accepts purchase orders as input and uses the Human Interaction Event service to store the purchase orders in the Application database and mark them as requiring approval. The Human Interaction Event service must also be configured to invoke a business process that constructs and sends to the approver an e-mail notification containing the URL for displaying the list of purchase orders. See the business scenario and business solution example for the B2B Lookup System service for an example business process.

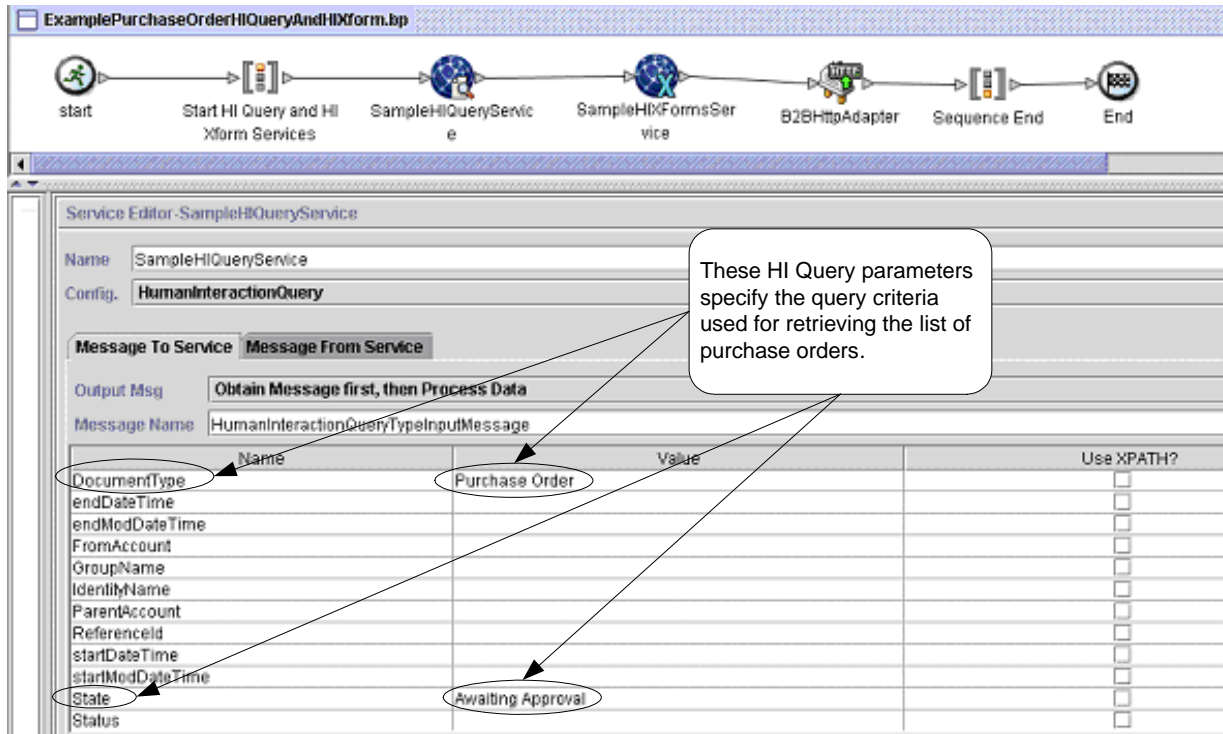
- Create a configuration of the HTTP Respond service that specifies the URI used to invoke (from the Web browser) the business process created in this example.

- Deploy a Web template in Application for displaying the list of purchase orders in the Web browser.

This business solution example focuses only on the Human Interaction Query service functionality.

GPM Example

The following GPM example shows a solution for retrieving the list of purchase orders requiring approval to display in a Web browser.



After the Human Interaction Query service completes, a list of purchase orders that are waiting for approval are returned to the business process in a document named `MinedDataQueryResults`. The returned document

for this example looks like the following, returning reference data for the two purchase orders meeting the query criteria:

```
<?xml version="1.0" encoding="utf-8"?>
<list xmlns="">
<listItem objectId="1446e4:fb36e3f660:1c5f"> ← Start of reference data for first purchase order
<referenceId>P01234</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner1</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>belushi:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/>
<fromMinedDataId>1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate><
marked/>
</listItem>
<listItem objectId="1446e4:fb36e3f660:1c67"> ← Start of reference data for second purchase order
<referenceId>P04567</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner2</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>belushi:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/>
<fromMinedDataId>1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate>
<marked/>
</listItem>
</list>
```

Note: The Human Interaction Document Loader service can use the value of the objectId attribute on the listItem element to retrieve the actual purchase order document associated with the reference data.

The MinedDataQueryResults document is then sent to the Human Interaction XForms service and the B2B HTTP Server adapter for display in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application

and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

For example, you might have a Web template that displays the reference data in the following format:

PURCHASE ORDER REQUESTS

<u>Order Number</u>	<u>Buyer</u>	<u>Request Date</u>	<u>State</u>
PO1234	TradingPartner1	03/20/2004	Awaiting Approval
PO4567	TradingPartner2	03/21/2004	Awaiting Approval

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML.

```
<process name="ExamplePurchaseOrderHIQueryAndHIXform">
<sequence name="Start HI Query and HI Xform Services">
<operation name="SampleHIQueryService">
<participant name="HumanInteractionQuery"/>
<output message="HumanInteractionQueryTypeInputMessage">
  <assign to="DocumentType">Purchase Order</assign>
  <assign to="State">Awaiting Approval</assign>
  <assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
<operation name="SampleHIXFormsService">
<participant name="HumanInteractionXForms"/>
<output message="HumanInteractionXFormsTypeInputMessage">
  <assign to="TemplateName"
from="DocToDOM(MinedDataQueryResults)/listItem/templateName/text()" />
  <assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
<operation name="HTTP Respond Service">
  <participant name="HttpRespond"/>
  <output message="HttpRespondServiceInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
```

```

        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Implementing the Human Interaction Query Service

To implement the Human Interaction Query service, complete the following tasks:

1. Activate your license for the Human Interaction Query service. See *Installing Application*.
2. Create a Human Interaction Query Service configuration. See *Managing Services and Adapters*.
3. Use the Human Interaction Query service in a business process.

Human Interaction Query Service Business Process Usage

The following screen shows a graphical view of the GPM parameters for the Human Interaction Query service. There are no fields to be configured on the Message From Service tab.

Name	Value	Use XPATH
DocumentType	Purchase Order	<input type="checkbox"/>
endDateTime	2004-02-25 00:00:00	<input type="checkbox"/>
endModDateTime	2004-02-25 00:00:00	<input type="checkbox"/>
FromAccount	SendingTradingPartnerSystemAccountID	<input type="checkbox"/>
GroupName	SendingTradingPartnerSecurityPermissionGroup	<input type="checkbox"/>
IdentityName	SendingTradingPartnerName	<input type="checkbox"/>
ParentAccount	ManagerOfTheDocumentSender	<input type="checkbox"/>
ReferenceId	PO1234	<input type="checkbox"/>
startDateTime	2004-02-24 00:00:00	<input type="checkbox"/>
startModDateTime	2004-02-24 00:00:00	<input type="checkbox"/>
State	Awaiting Approval	<input type="checkbox"/>
Status	Unread	<input type="checkbox"/>
StorageArea	Inbox	<input type="checkbox"/>
SystemAccount	ReceiverCompanyUserAccount	<input checked="" type="checkbox"/>

The following example shows the corresponding BPML parameters for the Human Interaction Query service GPM parameters.

```

<process name="ExamplePurchaseOrderHIQuery">
  <sequence name="Start HI Query Service">
    <operation name="SampleHIQueryBP">
      <participant name="HumanInteractionQuery"/>
      <output message="HumanInteractionQueryTypeInputMessage">
        <assign to="DocumentType">Purchase Order</assign>
        <assign to="endDateTime">2004-02-25 00:00:00</assign>
        <assign to="endModDateTime">2004-02-25 00:00:00</assign>
        <assign to="FromAccount">SendingTradingPartnerSystemAccountID</assign>
        <assign to="GroupName">SendingTradingPartnerSecurityPermissionGroup</assign>
      </output message>
    </operation>
  </sequence>
</process>

```



```

<assign to="IdentityName">SendingTradingPartnerName</assign>
<assign to="ParentAccount">ManagerOfTheDocumentSender</assign>
<assign to="ReferenceId">P01234</assign>
<assign to="startDateTime">2004-02-24 00:00:00</assign>
<assign to="startModDateTime">2004-02-24 00:00:00</assign>
<assign to="State">Awaiting Approval</assign>
<assign to="Status">Unread</assign>
<assign to="StorageArea">Inbox</assign>
<assign to="SystemAccount" from="ReceiverCompanyUserAccount"/>
<assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

The following table describes the fields used to configure the Human Interaction Query service in the GPM. The values specified for these parameters are the input to the Human Interaction Query service from the business process.

Note: If you do not specify any query criteria, no results are returned to the business process.

Field	Description
Config	Name of the service configuration.
DocumentType	Type of data that requires human interaction. Valid value is any alphanumeric string, such as Purchase Order, Invoice, and Remittance Advice.
endDateTime	End date and time of the time period for the Human Interaction Query service to search Application for original business processes flagged for human interaction. The format for this field is <i>yyyy-mm-dd hh:mm:ss</i> , with a space between the <i>dd</i> and <i>hh</i> . The default is to return the reference data for all documents meeting the specified query criteria.
endModDateTime	End date and time of the time period for the Human Interaction Query service to search Application for modified business processes flagged for human interaction. The format for this field is <i>yyyy-mm-dd hh:mm:ss</i> , with a space between the <i>dd</i> and <i>hh</i> . The default is to return the reference data for all documents meeting the specified query criteria.
FromAccount	Trading partner system user account ID (as defined in Application) of the trading partner that sent the document.
GroupName	Name of the security permission group to which the trading partner that sent the document belongs. Use this parameter to search records pertaining to a specific security permission group used for the Human Interaction Event service.
IdentityName	Name of the trading partner that sent the document. Use this parameter to search records pertaining to a specific company.
ParentAccount	User account name for the manager who sent the document. This user account name is associated with the FromAccount information in this service. Use this parameter to search records pertaining to a specific manager.

Field	Description
ReferenceID	Document identifier. Valid value is any alphanumeric string. Examples include a purchase order number or an invoice number.
startDateTime	Start date and time for the Human Interaction Query service to search Application for original documents flagged for human interaction. The format for this field is <i>yyyy-mm-dd hh:mm:ss</i> , with a space between the <i>dd</i> and <i>hh</i> . The default is to return the reference data for all documents meeting the specified query criteria.
StartModDateTime	Start date and time for the Human Interaction Query service to search Application for modified documents flagged for human interaction. The format for this field is <i>yyyy-mm-dd hh:mm:ss</i> , with space between the <i>dd</i> and <i>hh</i> . The default is to return the reference data for all documents meeting the specified query criteria.
State	State of the document in the process. Valid value is any alphanumeric string, such as Approved, Pending, or Rejected.
Status	Status of the document. Valid value is any alphanumeric string, such as Read or Unread.
StorageArea	Virtual location to store the data waiting for the person to take action. Valid value is any alphanumeric string, such as Inbox, Outbox, and Drafts.
SystemAccount	Trading partner user account ID (as defined in Application) of the person who must interact with the business process.

The following table describes additional fields used to configure the Human Interaction Query service. You can specify these fields using the Advanced Editor option in the GPM Service Editor or by using BPML.

Field	Description
Sort Order	Used to specify the Mined Data column that is used to sort the results returned by the Human Interaction Query service.
State2	An additional parameter that is only used by the *QuerySearch forms when the query includes more than just Active states, specifically when Active and Archive are the options. The Light Weight JDBC service can be use for queries that go beyond what the HI Query service can provide.

Output from Service to Business Process

The following table describes the output returned from the Human Interaction Query service to the business process:

Returned Output	Description
MinedDataQueryRes ults/@SCIOBJECTID document	The name of the document, along with the document SCI Object ID, returned by the Human Interaction Query service. This document contains the list of document reference data meeting the query criteria specified by the Human Interaction Query service parameters.

Human Interaction XForms Service

Use the Human Interaction XForms service to select the appropriate Web template for displaying business process data in a Web browser. The business process data could be documents and document reference data stored in the Application database or data returned from other services or adapters.

The Human Interaction XForms service:

Generates the HTML for the Web page based on the Web template and posts the HTML to the B2B HTTP Server adapter for display in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

Conducts data validation and calculations.

The following table provides an overview of the Human Interaction XForms service:

System name	HumanInteractionXForms
Graphical Process Modeler (GPM) categories	All Services, Web Extensions
Description	Directs Application to the appropriate Web template to use to display business process data. The Human Interaction XForms service displays the business process data in HTML format in a Web browser using a configured Web template.
Business usage	<p>A business process uses the Human Interaction Event service to save documents requiring approval in the database and then sends out an e-mail notification to the approver. The e-mail notification includes the URL for displaying the documents. When the approver specifies the URL in the browser, a business process runs that uses the Human Interaction Query service to retrieve all documents requiring approval from the database and uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the results in the Web browser for user selection.</p> <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See <i>Retiring and Removed Services and Adapters</i>.</p>

Usage example	<p>You can use the Human Interaction XForms service in various ways. For example:</p> <ul style="list-style-type: none"> ◆ The Human Interaction XForms service is preceded by a Human Interaction Query service that searches Application for business processes flagged for human interaction and saves the list in XML format. This list is passed to the Human Interaction XForms service, and the service identifies the Web template used to display the list. ◆ The Human Interaction XForms service is preceded by a Application service or adapter that produces XML as output and passes this output to the service for display in a Web browser. The Human Interaction XForms service identifies the Web template used to display the output. See <i>Usage Examples</i> for an example using this service to display data from the Lightweight JDBC adapter. <p>In each scenario, the Human Interaction XForms service is followed by the B2B HTTP Server adapter for displaying the information in the Web browser.</p> <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See <i>Retiring and Removed Services and Adapters</i>.</p>
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>Designed to work with:</p> <ul style="list-style-type: none"> ◆ Human Interaction Query service ◆ Human Interaction Event service ◆ B2B HTTP Server adapter ◆ Human Interaction Document Loader service <p>Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See <i>Retiring and Removed Services and Adapters</i>.</p>
Application requirements	None
Initiates business processes?	None
Invocation	Runs as part of a business process.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ WorkflowContext.SUCCESS: Success, with no errors. ◆ WorkflowContext.ERROR IllegalArgumentException: Caught IllegalArgumentException in XFormsService.processData() ◆ WorkflowContext.ERROR IllegalStateException: Caught IllegalStateException in XFormsService.processData() ◆ WorkflowContext.ERROR SQLException: Caught SQLException in XFormsService.processData()
Restrictions	None
Persistence level	None
Testing considerations	None

Requirements

Using this service requires:

- A valid, deployed Web template

- An enabled B2B HTTP Server adapter configuration

Note: The B2B HTTP Server adapter facilitates communications using the HTTP protocol. The B2B HTTP Server adapter must follow the Human Interaction XForms service for the document to be displayed in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

How the Human Interaction XForms Service Works

Use the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display business process data in a Web browser. The following sections describe a business scenario and a sample solution using the Human Interaction XForms service.

Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

Business Scenario

Your company receives purchase orders in XML format from a trading partner. All purchase orders exceeding \$1,000.00 need to be approved. You want to view a list of all purchase orders requiring approval in the Web browser.

The approach used to solve this business scenario includes creating a business process that includes:

- A Human Interaction Query service to retrieve the list of purchase orders requiring approval

- A Human Interaction XForms service to:

- ◆ Receive the list of purchase orders from the Human Interaction Query service.

Identify the Web template used to display the list.

- ◆ Forward the HTML from the Web template to the B2B HTTP Server adapter for displaying in a Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

The following dependencies must be met to run the example in this section and display the results in a Web browser:

Run a business process that accepts purchase orders as input and uses the Human Interaction Event service to store the purchase orders in the Application database and mark them as requiring approval. The Human Interaction Event service must also be configured to invoke a business process that constructs and sends an e-mail notification containing the URL for displaying the list of purchase orders to the approver. See the business scenario and business solution example for the Human Interaction Event service for an example business process.

Create a configuration of the HTTP Respond service that specifies the URI used to invoke the business process created in this example from the Web browser.

Deploy a Web template in Application for displaying the list of purchase orders in the Web browser.

This business solution example focuses only on the Human Interaction XForms service functionality.

GPM Example

The following example shows a solution to the business scenario using the GPM.

This parameter is set with the template name that was specified for the purchase order when it was added to the database. The HI XForms service retrieves this value from the MinedDataQueryResults document returned by the HI Query service. The MinedDataQueryResults document contains the reference data for the list of purchase orders requiring human interaction.

Name	Value	Use XPath?
ArchiveFlag		<input type="checkbox"/>
ArchiveHours		<input type="checkbox"/>
MergeDataOnResume		<input type="checkbox"/>
ServiceResponseMode		<input type="checkbox"/>
TemplateName	DocToDOM(MinedDataQueryResults\listitem\TemplateName.txt)	<input checked="" type="checkbox"/>

The Human Interaction Event service (not shown) generated an e-mail notification to the approver after the purchase orders were added to the database. The approver uses this URL to list the purchase orders requiring approval in the Web browser. For example, `http://siHostName:siport/webx/bp/businessProcessName`, where *businessProcessName* is the name of the business process to invoke.

The business process that runs uses the Human Interaction Query service to retrieve the list of purchase orders from the database and uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the list.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

After the Human Interaction Query service completes, a list of purchase orders that are waiting for approval are returned to the business process in a document named MinedDataQueryResults. The returned document for this example looks like the following, returning reference data for the two purchase orders:

```

<?xml version="1.0" encoding="utf-8"?>
<list xmlns="">
<listItem objectId="1446e4:fb36e3f660:1c5f">
<referenceId>PO1234</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner1</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>server1:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/><fromMinedDataId>
1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate>
<marked/>
</listItem>
<listItem objectId="1446e4:fb36e3f660:1c67">
<referenceId>PO4567</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner2</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>server1:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/>
<fromMinedDataId>1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate>
<marked/>
</listItem>
</list>

```

Web template used for displaying the data

The MinedDataQueryResults document is then sent to the Human Interaction XForms service, which identifies the Web template used for displaying the list of purchase orders and returns the HTML to the business process. The Human Interaction XForms service forwards the HTML to the B2B HTTP Server adapter for display in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

For example, you might have a Web template that displays the reference data in the following HTML format:

PURCHASE ORDER REQUESTS

<u>Order Number</u>	<u>Buyer</u>	<u>Request Date</u>	<u>State</u>
PO1234	TradingPartner1	03/20/2004	Awaiting Approval
PO4567	TradingPartner2	03/21/2004	Awaiting Approval

The approver can select the purchase order number from the list to display the purchase order in the Web template and then approve or reject the order. See Human Interaction Document Loader service for information about retrieving a specific document from the database for display purposes.

The approver could alternatively log in to the Application Web Suite application to view (in their inbox) a list of documents requiring approval.

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML:

```
<process name="ExamplePurchaseOrderHIQueryAndHIXform">
<sequence name="Start HI Query and HI Xform Services">
<operation name="SampleHIQueryService">
<participant name="HumanInteractionQuery"/>
<output message="HumanInteractionQueryTypeInputMessage">
  <assign to="DocumentType">Purchase Order</assign>
  <assign to="State">Awaiting Approval</assign>
  <assign to="SystemAccount" from="system-account-user-id/text()"/>
  <assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
<operation name="SampleHIXFormsService">
<participant name="HumanInteractionXForms"/>
<output message="HumanInteractionXFormsTypeInputMessage">
  <assign to="TemplateName"
from="DocToDOM(MinedDataQueryResults)/listItem/templateName/text()"/>
  <assign to="." from="*" />
</output>
<input message="inmsg">
```

```

    <assign to="." from="*" />
</input>
</operation>
<operation name="HTTP Respond Service">
  <participant name="HttpRespond" />
  <output message="HttpRespondServiceInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</process>

```

See *Usage Examples* for an additional example of using the Human Interaction XForms service.

Implementing the Human Interaction XForms Service

To implement the Human Interaction XForms service, complete the following tasks:

1. Activate your license for the Human Interaction XForms service. See *Installing Application*.
2. Create a Human Interaction XForms service configuration. See *Managing Services and Adapters*.
3. Use the Human Interaction XForms service in a business process.

Human Interaction XForms Service Business Process Usage

The following screen shows a graphical view of the GPM parameters for the Human Interaction XForms service. There are no fields to be configured on the Message From Service tab.

Name	Value	Use XPath?
ArchiveFlag	1	<input type="checkbox"/>
ArchiveHours	24	<input type="checkbox"/>
MergeDataOnResume	No	<input type="checkbox"/>
SerialResponseMode	No	<input type="checkbox"/>
TemplateName	WebTemplateName	<input type="checkbox"/>

The following example shows the corresponding BPML parameters for the Human Interaction XForms service GPM parameters.

```

<process name="ExampleHIXFormsGPMParms">
<sequence name="Start">
<operation name="SampleHIXFormsService">
<participant name="HumanInteractionXForms" />
<output message="HumanInteractionXFormsTypeInputMessage">

```

```

    <assign to="ArchiveFlag">1</assign>
    <assign to="ArchiveHours">24</assign>
    <assign to="MergeDataOnResume">NO</assign>
    <assign to="ServletResponseMode">NO</assign>
    <assign to="TemplateName">WebTemplateName</assign>
    <assign to="." from="*" />
</output>
<input message="inmsg">
    <assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

The following table describes the fields used to configure the Human Interaction XForms service in the GPM. The values specified for these parameters are the input to the Human Interaction XForms service from the business process.

Field	Description
Config	Name of the service configuration.
ArchiveHours	The number of hours the mined data record should remain in the Web Extension table before it is archived or purged. Valid value is a positive integer. Optional. If blank, the time span from the business process is used, if it has been specified. If the time span from the business process has not been specified, the system default is used.
ArchiveFlag	The archive method that should be used. Optional. Valid values: <ul style="list-style-type: none"> ◆ 1 – Archives the data in the Web Extension table. ◆ 2 – Deletes the data from the Web Extension table. ◆ blank – The archive method from the business process is used.
MergeDataOnResume	Merges any changes made to the Web template with the existing instance data when the business process is taken out of the waiting state. Valid values: <ul style="list-style-type: none"> ◆ Yes – Causes the documents and instance data to be merged when the business process is resumed. ◆ No – Overlays the business process being resumed (taken out of a waiting state) with the changed documents and instance data. Default.

Field	Description
ServletResponseMode	<p>Determines the operation of the Human Interaction XForms service.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Recognizes that the business process started by a Business Process XLink Resolver. The Human Interaction XForms service disregards all other configured parameters and immediately sends the primary document back to the XForms servlet for processing. ◆ No – Parameter has no affect. The service works as configured. Default. <p>Note: The Business Process XLink Resolver is a Java class that starts business processes and is used in place of the HTTP Servlet adapter for Web Extensions running on Application. In this case, business processes start directly by a call to the workflow class.</p>
TemplateName	<p>Name of the Web template to use for viewing the data. Select the template name from the list of deployed templates, or dynamically select the template name from the query results returned by the Human Interaction Query service.</p>

Usage Examples

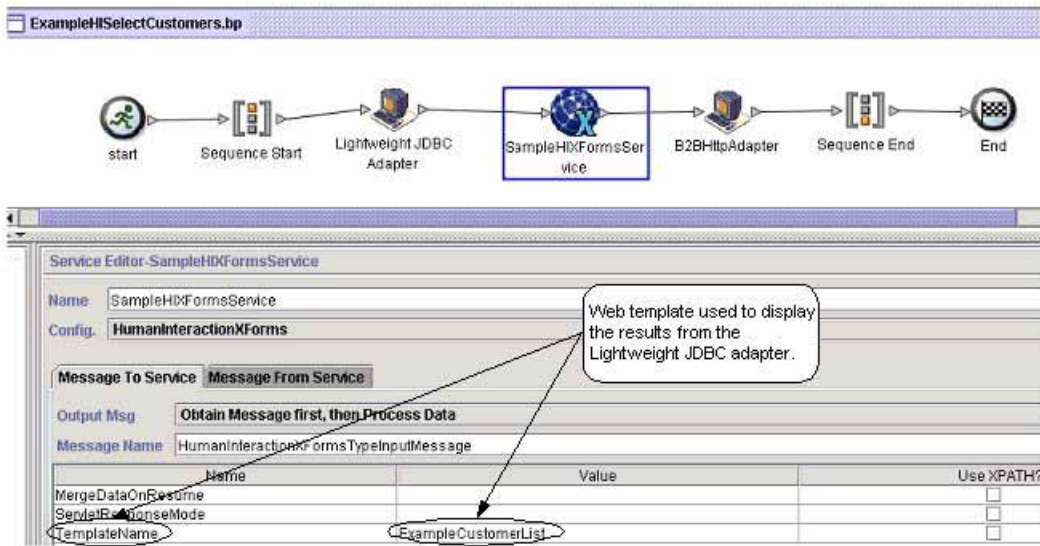
The example in this section illustrates a business process that uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the results returned from a Lightweight JDBC adapter. The Lightweight JDBC adapter is used to query the Application database for customer records and return the results as a primary document to the business process. The primary document is passed to the Human Interaction XForms service, which identifies the Web template and generates the HTML for displaying the customer information. The Human Interaction XForms service then posts the document to the B2B HTTP Server adapter for display in the Web browser.

Caution: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Application and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

This business process started by specifying a URL in the Web browser (for example, `http://siHostName:siPort/webx/bp/ExampleCustomerList`), which is accomplished with the Lightweight JDBC adapter.

Note: This example assumes that there is a valid, deployed Web template in Application for displaying the list of customers in the Web browser and that there is URI mapping defined in the B2B HTTP Server adapter configuration for invoking the business process from an HTTP request.

The following example illustrates this scenario using the GPM.



The Lightweight JDBC adapter returns to the business process the following customer list as the primary document. This document is input to the Human Interaction XForms service.

```
<?xml version='1.0' encoding='UTF-8'?>
<CustomerList>
<RowsReturnedFromDatabase>
<customer_id>1234</customer_id>
<customer_name>John Doe</customer_name>
<customer_address>address1</customer_address>
<customer_phone>xxxx-xxx-xxxx</customer_phone>
</RowsReturnedFromDatabase>
<RowsReturnedFromDatabase>
<customer_id>5678</customer_id>
<customer_name>Jane Doe</customer_name>
<customer_address>address1</customer_address>
<customer_phone> xxxx-xxx-xxxx</customer_phone>
</RowsReturnedFromDatabase>
</CustomerList>
```

The user specifies the URL in the Web browser for example, <http://siHostName:siPort/webx/bp/ExampleCustomerList>, which invokes the ExampleHISelectCustomers

business process and displays the list of customers in the Web browser. For example, you might have a Web template that displays the list of customers in the following HTML format:

CUSTOMER LIST

<u>Customer ID</u>	<u>Name</u>	<u>Address</u>	<u>Phone</u>
1234	John Doe	address1	xxx-xxx-xxxx
4567	Jane Doe	address2	xxx-xxx-xxxx

The following example illustrates the business process using BPML:

```

<process name="ExampleHISelectCustomers">
<sequence>
<operation name="Lightweight JDBC Adapter">
  <participant name="ExampleLWJDBCBusinessProcess"/>
  <output message="LightweightJDBCAdapterTypeInputMessage">
    <assign to="pool">mysqlTrainingPool</assign>
    <assign to="query_type">SELECT</assign>
    <assign to="result_name">CustomerList</assign>
    <assign to="row_name">RowsReturnedFromDatabase</assign>
    <assign to="sql">SELECT * FROM Customer</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="Document" from="PrimaryDocument/@SCIOBJECTID"/>
  </input>
</operation>
<operation name="SampleHIXFormsService">
  <participant name="HumanInteractionXForms"/>
  <output message="HumanInteractionXFormsTypeInputMessage">
    <assign to="TemplateName">ExampleCustomerList</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
<operation name="HTTP Respond Service">
  <participant name="HttpRespond"/>
  <output message="HttpRespondServiceInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>

```

</process>

IBM Information Exchange FTP (IBM IE FTP) Adapter

The following table provides an overview of the IBM Information Exchange FTP (IBM IE FTP) adapter, which is used to communicate with the IBM Information Exchange network:

System name	IBM IE FTP adapter
Graphical Process Modeler (GPM) category	All Services
Description	The IBM IE FTP adapter is used to send and collect documents from the IBM EDI services mailbox using a TCP/IP FTP gateway.
Business usage	Used to exchange documents with a trading partner's IBM EDI services mailbox.
Usage example	An internal service invokes a business process that sends a trading document to a trading partner mailbox in the IBM VAN.
Preconfigured?	No
Requires third party files?	None
Platform availability	All supported Application platforms
Related adapters	FTP Client Adapter
Application requirements	None
Initiates business processes?	Yes, this adapter will initiate another business process in COLLECT mode.
Invocation	This adapter can be invoked by an internal service only.
Business process context considerations	You must be familiar with the internal service (workflow and business process parameters) that invoked this adapter. Workflow parameters are those values that were passed into the internal service. Business process parameters are those values specified within the business process code.
Returned status values	<ul style="list-style-type: none">◆ 0 – Success◆ 1 – Error
Restrictions	None
Persistence level	System default (Full)

Testing considerations	<p>Create a test configuration of the adapter and create a business process to test with. See <i>Business Process Example</i> on page 693.</p> <p>The most frequent problems encountered are:</p> <ul style="list-style-type: none"> ◆ Parameters are configured incorrectly. ◆ Adapter is not active. ◆ Response timeout. The IBM server may not always return the status code of the submission before the adapter timeout value is reached. When this happens, the business process is halted and a response timeout message is written to the Advanced Status. This may not mean that the submission has failed, however, because the IBM Server would have processed all the documents and sent them to their respective mailboxes before attempting to send back the response. When writing an On Error script, have the script check for the error message before deciding on whether to retry the submission. Also, set a long timeout value. Or, if possible, check with the trading partner to see if the document has been received.
------------------------	--

Implementing the IBM IE FTP Adapter

To implement the IBM IE FTP adapter, complete the following tasks:

1. Activate your license for the IBM IE FTP adapter. See *An Overview of Implementing Services*.
2. Create an IBM IE FTP adapter configuration. See *Creating a Service Configuration*.
3. Configure the adapter. See *Configuring the IBM IE FTP Adapter* on page 690.
4. Create a business process that includes the IBM IE FTP adapter and enable it.
5. Test the business process and the adapter.
6. Run the business process.

Configuring the IBM IE FTP Adapter

To configure the IBM IE FTP adapter, you must specify field settings in Application.

Application Configuration

The following table describes the fields used to configure the IBM IE FTP adapter in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time (default) ◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>See <i>Using Service Groups</i>.</p>
Default IBM IE FTP Server	The host information for IBM IE FTP server. Valid value is a hostname or IP address. Default is myhost. Required.
Default IBM IE FTP Server Port	Port number of the IBM IE FTP server. This is typically port number 21, but it can be another number depending on the FTP server settings. Default is 21. Required.
User Name	User login name to connect to the IBM IE FTP server. Alphanumeric and case-sensitive. Default is euser. Required.
Password	Password used for authentication when connecting to the IBM IE FTP server. Case-sensitive. Default is *****. Required.
SMIME Encryption User Certificate	Not in use
SMIME Decryption Certificate (System Store):	Not in use
SSL	<p>The SSL flag that determines SSL socket negotiation. Valid values are:</p> <ul style="list-style-type: none"> ◆ SSL_IMPLICIT – Indicates that the FTP server expects and requires SSL to happen automatically at the time of connection. ◆ SSL_EXPLICIT – Indicates that the FTP client requests SSL and a secure connection is negotiated. ◆ SSL_NONE – Indicates that the connection will not use SSL. <p>Required.</p>
Clear Control Channel	Indicates if information that travels across the control channel should be clear. Valid values are Yes and No. Default is No. Required.
Key Certificate Passphrase	Not in use.

Field	Description
Cipher Strength	<p>The level of encryption you would like to be applied to the data that flows through the socket connection. Valid values are:</p> <ul style="list-style-type: none"> ◆ ALL ◆ WEAK ◆ STRONG (default) <p>Optional.</p>
Key Certificate (System Store)	<p>Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner trusted certificate authority. In process data, this parameter is displayed as an object ID. Required for client authentication.</p>
CA Certificates	<p>Select from the list of trusted certificate authority public certificates. In process data, this parameter is displayed as an object ID. Required for server authentication.</p>
Select the operation	<p>Specifies the type of operation to perform. Valid values are:</p> <ul style="list-style-type: none"> ◆ COLLECT (default) ◆ SUBMIT <p>Required.</p>
Trading Partner	<p>Specifies the trading partner mailbox to collect. Default is account.userid. Required for COLLECT.</p>
Message Class	<p>Specifies the message class to collect. Default is text. Required for COLLECT.</p>
Business Process	<p>Specifies the business process to invoke. Required for COLLECT.</p>
Schedule Settings	<p>Enables you to set up a schedule. Supporting scheduling options are presented based on the value specified here. Valid values are:</p> <ul style="list-style-type: none"> ◆ Do not use schedule ◆ Run based on timer (default) ◆ Run daily ◆ Run based on day(s) of the week ◆ Run based on day(s) of the month <p>Required for COLLECT.</p>
Select the message type	<p>Specifies the message type to send. If you select SUBMIT for the type of operation, you will be presented with this option. Valid values are:</p> <ul style="list-style-type: none"> ◆ EDI message (default) ◆ Non EDI message <p>Required for SUBMIT.</p>

Field	Description
Select the alias table type	<p>Specifies the alias table type. If you select EDI for the message type, this field is displayed. You define the alias table to resolve EDI addresses. Valid values are:</p> <ul style="list-style-type: none"> ◆ Private alias table – can only be accessed by the user (default). ◆ Global alias table – can be accessed by all users in the system. ◆ Organizational alias table – can be accessed by all users within the same organization. <p>Required for SUBMIT EDI message.</p>
Alias table name	Specifies the alias table name. Default is tablename. Required for SUBMIT EDI message.

Business Process Example

The following business process example illustrates how you can invoke an IBM IE FTP adapter configuration:

```

<process name="Test_bpml">
  <sequence name="optional">
    <operation name="Invoke an instance of IBM IE FTP adapter ">
      <participant name="IBM_IE_FTP_adapter"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Import Service (Build 4300 - Build 4321)

The following table provides an overview of the Import service:

System name	ImportService
Graphical Process Modeler (GPM) category	All Services
Description	<p>This service is used in a business process to automatically import application resources exported using the Resource Manager, including:</p> <ul style="list-style-type: none">◆ SAP application configurations◆ Translation maps◆ Trading partner data (packages, identities, contracts, envelopers, and code lists)◆ Business processes◆ Service configurations◆ XML schemas◆ XSLT stylesheets◆ Web templates◆ Web resources (JSP files, JavaScript files, HTML files, XML files, image files, property files, stylesheets, and custom defined files)
Business usage	In a hub and spoke relationship, a hub company could use this service to programatically update information on their trading partners' systems.
Usage example	A hub needs to update its trading partner information with all of its spokes. At the same time, it plans on rolling out new XML schemas and translation maps. The hub creates an installable bundle interactively by means of the resource manager. The bundle is sent to the trading partners affected. The bundle is picked up by the trading partners and processed by a business process set up as the updates from trading partner hub x. This process includes the Import service. The service checks the security context and, assuming it is correct, opens the bundle and updates the local system with the updates automatically.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported application platforms
Related services	This service is designed to work in conjunction with a transport type service. The transport service brings the resources into the local system.
Application requirements	Before using this service, a security context for the installable resource bundle must have been created using the Security Context utility.
Initiates business processes?	No
Invocation	Event driven.

Business process context considerations	The Import service configuration may contain context and identity values for a security context, which are used during the business process execution to fetch the passphrase for verification if the file being imported contains encrypted data. If the passphrase is required but the values are not for the correct passphrase or no security context information is available (either the values from the service configuration or the passphrase value stored for each configuration in the application database), the Import service will fail.
Returned status values	<ul style="list-style-type: none"> ◆ Success – The service completed successfully. ◆ Error – The service experienced a fatal error while processing.
Restrictions	<ul style="list-style-type: none"> ◆ This service does not construct installable bundles or export resources. These operations must be done interactively. ◆ All resources defined in the installable bundle will be installed. ◆ Any existing resources will be updated, and the version number incremented. ◆ The installed resource will become the default, if applicable.
Persistence level	Full
Testing considerations	Export a set of resources from application to a file called Export.xml. Import these resources in to another application server. Check the status report. There must not be any errors and it should be possible to test the imported resources.

How the Import Service Works

The Import service exercises the same functionality as the Import Resources option in the Resource Manager, with one exception—the service has no user interaction, so the service does not ask for confirmation of options. When using the Import service, all available resources are imported, and all imported versions are set as the defaults, where applicable.

You create a security context for an installable bundle, which can prevent unauthorized users from creating or updating resources.

The Import service works with the Security Context utility in the application. The utility is called `securityContext.sh` (for Unix) or `securityContext.cmd` (for Windows). It is located in the bin directory of your application installation. This is an example of how the security context is used:

1. A developer at company A exports a resource bundle to be sent to company B, where the bundle will be imported. If required for the type of resource to be exported, the developer creates a passphrase for the resource bundle as a part of the export process.
2. After the export is complete, the developer sends the resource bundle to the company B system administrator, and also informs the system administrator of the passphrase.
3. The system administrator at company B uses the Security Context utility to enter the passphrase into the application database and to create a security context.
4. The system administrator passes the resource bundle and the name of the security context to a developer.
5. The developer configures the Import service in the GPM, using the context and identity values from the security context that the system administrator provided.

For more information about creating a security context, see *Using the Security Context Utility*.

Implementing the Import Service

To implement the Import service, complete the following tasks:

1. After receiving a resource bundle from a trading partner, create a security context for it. For information, see *Using the Security Context Utility*.
2. Create an Import service configuration. See *Managing Services and Adapters*.
3. Configure the Import service.
4. Use the Import service in a business process.

Configuring the Import Service Configuration

To configure the Import service, you must specify settings for the following fields in application:

Field	Description
Config	Name of the service configuration.
Context	The company from which the resource files are obtained. Required if the file to be imported contains encrypted data; otherwise, optional. Example: Company_x
Identity	An ID to identify various passphrases received on various dates from the same company. Required if the file to be imported contains encrypted data; otherwise, optional. Example: 10Jan2004
KeepExistingControl Numbers	Specifies whether the control numbers in the import file will be imported. The default for this parameter is No , which specifies that the control numbers in the import file will be imported. If you change this parameter to Yes , it specifies that for existing envelopes and control numbers, control number values in the import file will not be imported. If a version of an envelope or control number being imported already exists in the system, the import process will overwrite the value specified in the import file with the existing control number value for that envelope or control number.

Output from Business Process to Service

The following table describes the output from the business process to the Import service:

Parameter	Description
Filename	The name of the resource file, including full path information. Valid value is any valid path and filename.

Using the Security Context Utility

There are three actions you can perform with the security context command: list, get, and set. The security context command file is located in the bin directory of your application installation.

Action	Description	Usage
list	Lists all security contexts available.	Unix: <i>install_dir/bin>securityContext.sh list_context</i> Windows: <i>install_dir\bin>securityContext.cmd list_context</i>
set	Updates the database in application with the new context. Takes three parameters: <ul style="list-style-type: none">◆ context◆ identity◆ passphrase	Unix: <i>install_dir/bin>securityContext.sh set context identity passphrase</i> Windows: <i>install_dir\bin>securityContext.cmd set context identity passphrase</i> Returns the following message: Context saved.
get	Returns the passphrase value for the context. Takes two parameters: context and identity.	Unix: <i>install_dir/bin>securityContext.sh get context identity</i> Windows: <i>install_dir\bin>securityContext.cmd get context identity</i> Returns the following values: <i>context, identity, password</i>

Example

In the following example, the Kimata company's system administrator, Jill, creates a security context called MaxxMart for an exported resource bundle just received from their trading partner, MaxxMart. Jill sets the identity for this context to dec19 (date it was received from the trading partner). MaxxMart also sent Jill the passphrase that they created for the resource bundle: bubblegum.

```
install_dir\bin>securityContext.sh set MaxxMart dec19 bubblegum
```

In the second example, Jill wants to find out what security contexts are on her application system, and uses the list_context action to find out. There are three contexts on the system: MaxxMart, Taylor, and Zapf.

```
install_dir\bin>securityContext.sh list_context  
Contexts:  
MaxxMart, Taylor, Zapf
```

In the third example, Jill wants to know what the passphrase is for the security context named Taylor that has an identity of jan20. She uses the get action and finds that the passphrase is thunder.

```
install_dir\bin>securityContext.sh get Taylor jan20  
Taylor, jan20, thunder
```

Business Process Example 1

The following example illustrates using the Import service in a business process to import a resource file called dec19 from Company_x:

```
<process name="ImportService">
  <sequence>
    <operation>
      <participant name="ImportService"/>
      <output message="Xout">
<assign to="Context">company_x</assign>
<assign to="Identity">dec19</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
    </operation>
  </sequence>
</process>
```

Business Process Example 2

The following example illustrates using the Import service in a business process to import a resource file called april1 from RomansFloorsAndMore, using the Keep Existing Control Numbers option (specifying that for existing envelopes and control numbers, control number values in the import file will not be imported, and i) if a version of an envelope or control number being imported already exists in the system, the import process will overwrite the value specified in the import file with the existing control number value for that envelope or control number:

```
<process name="ImportServiceWithKeepExistingControlNumbers">
<sequence>
  <operation>
    <participant name="ImportService"/>
    <output message="Xout">
      <assign to="Context">RomansFloorsAndMore</assign>
      <assign to="Identity">april1</assign>
      <assign to="KeepExistingControlNumbers">True</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
    </operation>
  </sequence>
</process>
```

Viewing the Import Service Status Report

Once you have imported resources using the Import service in a business process, it is a good idea to check that all resources were imported successfully. Also, you may need to use the report for troubleshooting if the service and business process fail. You can view the status report from the Business Process Detail page.

To view the report Business Process Detail page, complete the following steps:

1. From the Business Process menu, select **Monitor > Current Processes**. Current business processes are displayed in a list.
2. Click the Instance ID next to the desired business process. The Business Process Detail page for that business process displays.
3. In the Status Report column, click the Info icon for the Import service. The report is opened in another window. The status of each resource that the service attempted to import is shown, which enables you to verify whether each was successfully imported or not.

Some reasons that the Import service might fail (which will cause the business process to fail) in situations where passphrase (Context/Identity) is required are:

Invalid passphrase (the passphrase in the database doesn't match the passphrase in the resource bundle).

No passphrase in situation where passphrase is required (possibly there was no security context created for this resource bundle).

Either the context or identity value in the Import service configuration used in the business process is wrong or was left blank.

The following is a sample status report for an import bundle. There is one error for a resource that could not be imported (transport account password):

```
Name: UpdateTPInfo      Instance ID:1053      Service Name: Import Service
  Status report on 2004-03-12 14:45:19.16 for service: Import
Packaging :: packaging_1079119091618 :: update :: SUCCESS :: Resource successfully
imported.
Identity :: MaxxMart :: update :: SUCCESS :: Resource successfully imported.
Transport :: HTTP Transport :: create :: Message :: Error decrypting transport
account password...value will be stored as it was in import file.
Transport :: HTTP Transport :: update :: SUCCESS :: Resource successfully imported.
Document Exchange :: MaxxMart Doc Exchange :: update :: SUCCESS :: Resource
successfully imported.
Delivery Channel :: ABCD :: update :: SUCCESS :: Resource successfully imported.
Profile :: MaxxMart1 :: update :: SUCCESS :: Resource successfully imported.
End of report
ImportService stayed in queue 7 ms
```

Import Service (Build 4322 or higher)

The following table provides an overview of the Import service:

System name	ImportService
Graphical Process Modeler (GPM) category	All Services
Description	<p>This service is used in a business process to automatically import application resources exported using the Resource Manager, including:</p> <ul style="list-style-type: none">◆ SAP application configurations◆ Translation maps◆ Trading partner data (packages, identities, contracts, envelopers, and code lists)◆ Business processes◆ Service configurations◆ XML schemas◆ XSLT stylesheets◆ Web templates◆ Web resources (JSP files, JavaScript files, HTML files, XML files, image files, property files, stylesheets, and custom defined files)
Business usage	In a hub and spoke relationship, a hub company could use this service to programatically update information on their trading partners' systems.
Usage example	A hub needs to update its trading partner information with all of its spokes. At the same time, it plans on rolling out new XML schemas and translation maps. The hub creates an installable bundle interactively by means of the resource manager. The bundle is sent to the trading partners affected. The bundle is picked up by the trading partners and processed by a business process set up as the updates from trading partner hub x. This process includes the Import service. The service checks the security context and, assuming it is correct, opens the bundle and updates the local system with the updates automatically.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms
Related services	This service is designed to work in conjunction with a transport type service. The transport service brings the resources into the local system.
Application requirements	Before using this service, a security context for the installable resource bundle must have been created using the Security Context utility.
Initiates business processes?	No
Invocation	Event driven.

Business process context considerations	The Import service configuration may contain context and identity values for a security context, which are used during the business process to fetch the passphrase for verification if the file being imported contains encrypted data. If a passphrase is required but the values are not for the correct passphrase or no security context information is available (either the values from the service configuration or the passphrase value stored for each configuration in the application database), the Import service will fail.
Returned status values	<ul style="list-style-type: none"> ◆ Success – The service completed successfully. ◆ Error – The service experienced a fatal error while processing.
Restrictions	<ul style="list-style-type: none"> ◆ This service does not construct installable bundles or export resources. These operations must be done interactively. ◆ All resources defined in the installable bundle will be installed. ◆ Any existing resources will be updated, and the version number incremented. ◆ The installed resource will become the default, if applicable.
Persistence level	Full
Testing considerations	Export a set of resources from the application to a file called Export.xml. Import these resources in to another application server. Check the status report. There must not be any errors and it should be possible to test the imported resources.

How the Import Service Works

The Import service exercises the same functionality as the Import Resources option in the Resource Manager, with one exception—the service has no user interaction, so the service does not ask for confirmation of options. When using the Import Service, all available resources are imported, and all imported versions are set as the defaults, where applicable.

You can create a security context for an installable bundle, which can prevent unauthorized users from creating or updating resources.

The Import Service works with the Security Context utility in the application. The utility is called `securityContext.sh` (for Unix) or `securityContext.cmd` (for Windows). It is located in the bin directory of your application installation. This is an example of how the security context is used:

1. A developer at company A exports a resource bundle to be sent to company B, where the bundle will be imported. If required for the type of resource to be exported, the developer creates a passphrase for the resource bundle as a part of the export process.
2. After the export is complete, the developer sends the resource bundle to the company B system administrator, and also informs the system administrator of the passphrase.
3. The system administrator at company B uses the Security Context utility to enter the passphrase into the application database and to create a security context.
4. The system administrator passes the resource bundle and the name of the security context to a developer.
5. The developer configures the Import service in the GPM, using the context and identity values from the security context that the system administrator provided.

For more information about creating a security context, see *Using the Security Context Utility*.

Implementing the Import Service

To implement the Import Service, complete the following tasks:

1. After receiving a resource bundle from a trading partner, create a security context for it. For information, see *Using the Security Context Utility*.
2. Create an Import Service configuration. See *Managing Services and Adapters*.
3. Configure the Import Service.
4. Use the Import Service in a business process.

Configuring the Import Service

To configure the Import Service, you must specify settings for the following fields in your application:

Field	Description
Backup	Identifies the path where the backup is saved. If the path is invalid during backup, the file is written to <install>/tmp and a message is added to the Import Report indicating the location. If the parameter is not specified, then the backup is not generated.
Config	Name of the service configuration.
Context	The company from which the resource files are obtained. Required if the file to be imported contains encrypted data; otherwise, optional. Example: Company_x
Identity	An ID to identify various passphrases received on various dates from the same company. Required if the file to be imported contains encrypted data; otherwise, optional. Example: 10Jan2004
KeepExistingControl Numbers	Specifies whether the control numbers in the import file will be imported. The default for this parameter is No , which specifies that the control numbers in the import file will be imported. If you change this parameter to Yes , it specifies that for existing envelopes and control numbers, control number values in the import file will not be imported. If a version of an envelope or control number being imported already exists in the system, the import process will overwrite the value specified in the import file with the existing control number value for that envelope or control number.

Output from Business Process to Service

The following table describes the output from the business process to the Import Service:

Parameter	Description
Filename	The name of the resource file, including full path information. Valid value is any valid path and filename.

Using the Security Context Utility

There are three actions you can perform with the security context command: list, get, and set. The security context command file is located in the bin directory of your application installation.

Action	Description	Usage
list	Lists all security contexts available.	Unix: <i>install_dir/bin>securityContext.sh list_context</i> Windows: <i>install_dir\bin>securityContext.cmd list_context</i>
set	Updates the database in application with the new context. Takes three parameters: <ul style="list-style-type: none">◆ context◆ identity◆ passphrase	Unix: <i>install_dir/bin>securityContext.sh set context identity passphrase</i> Windows: <i>install_dir\bin>securityContext.cmd set context identity passphrase</i> Returns the following message: Context saved.
get	Returns the passphrase value for the context. Takes two parameters: context and identity.	Unix: <i>install_dir/bin>securityContext.sh get context identity</i> Windows: <i>install_dir\bin>securityContext.cmd get context identity</i> Returns the following values: <i>context, identity, password</i>

Example

In the following example, the Kimata company's system administrator, Jill, creates a security context called MaxxMart for an exported resource bundle just received from their trading partner, MaxxMart. Jill sets the identity for this context to dec19 (date it was received from the trading partner). MaxxMart also sent Jill the passphrase that they created for the resource bundle: bubblegum.

```
install_dir\bin>securityContext.sh set MaxxMart dec19 bubblegum
```

In the second example, Jill wants to find out what security contexts are on her application system, and uses the list_context action to find out. There are three contexts on the system: MaxxMart, Taylor, and Zapf.

```
install_dir\bin>securityContext.sh list_context
Contexts:
MaxxMart, Taylor, Zapf
```

In the third example, Jill wants to know what the passphrase is for the security context named Taylor that has an identity of jan20. She uses the get action and finds that the passphrase is thunder.

```
install_dir\bin>securityContext.sh get Taylor jan20
Taylor, jan20, thunder
```

Business Process Example 1

The following example illustrates using the Import service in a business process to import a resource file called dec19 from Company_x:

```
<process name="ImportService">
  <sequence>
    <operation>
      <participant name="ImportService"/>
      <output message="Xout">
<assign to="Context">company_x</assign>
<assign to="Identity">dec19</assign>
      <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
      <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Business Process Example 2

The following example illustrates using the Import service in a business process to import a resource file called april1 from RomansFloorsAndMore, using the Keep Existing Control Numbers option (specifying that for existing envelopes and control numbers, control number values in the import file will not be imported, and if a version of an envelope or control number being imported already exists in the system, the import process will overwrite the value specified in the import file with the existing control number value for that envelope or control number:

```
<process name="ImportServiceWithKeepExistingControlNumbers">
<sequence>
  <operation>
    <participant name="ImportService"/>
    <output message="Xout">
      <assign to="Context">RomansFloorsAndMore</assign>
      <assign to="Identity">april1</assign>
      <assign to="KeepExistingControlNumbers">True</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
</process>
```

Viewing the Import Service Status Report

Once you have imported resources using the Import service in a business process, it is a good idea to check that all resources were imported successfully. Also, you may need to use the report for troubleshooting if the service and business process fail. You can view the status report from the Business Process Detail page.

To view the report Business Process Detail page, complete the following steps:

1. From the Business Process menu, select **Monitor > Current Processes**. Current business processes are displayed in a list.
2. Click the Instance ID next to the desired business process. The Business Process Detail page for that business process displays.
3. In the Status Report column, click the Info icon for the Import service. The report is opened in another window. The status of each resource that the service attempted to import is shown, which enables you to verify whether each was successfully imported or not.

Some reasons that the Import service might fail (which will cause the business process to fail) in situations where passphrase (Context/Identity) is required are:

Invalid passphrase (the passphrase in the database doesn't match the passphrase in the resource bundle).

No passphrase in situation where passphrase is required (possibly there was no security context created for this resource bundle).

Either the context or identity value in the Import service configuration used in the business process is wrong or was left blank.

The following is a sample status report for an import bundle. There is one error for a resource that could not be imported (transport account password):

```
Name: UpdateTPInfo      Instance ID:1053      Service Name: Import Service
  Status report on 2004-03-12 14:45:19.16 for service: Import
Packaging :: packaging_1079119091618 :: update :: SUCCESS :: Resource successfully
imported.
Identity :: MaxxMart :: update :: SUCCESS :: Resource successfully imported.
Transport :: HTTP Transport :: create :: Message :: Error decrypting transport
account password...value will be stored as it was in import file.
Transport :: HTTP Transport :: update :: SUCCESS :: Resource successfully imported.
Document Exchange :: MaxxMart Doc Exchange :: update :: SUCCESS :: Resource
successfully imported.
Delivery Channel :: ABCD :: update :: SUCCESS :: Resource successfully imported.
Profile :: MaxxMart1 :: update :: SUCCESS :: Resource successfully imported.
End of report
ImportService stayed in queue 7 ms
```

Index Business Process Service

The Index Business Process service sets all completed or terminated business process data to be ready for archiving or purging. Creates an entry for each completed or terminated business process in the WF_INST_S table, and updates several tables. The following table provides an overview of the Index Business Process service:

System name	Index_Service
Graphical Process Modeler (GPM) categories	All Services, System
Description	Sets all completed or terminated business process data to be ready for archiving or purging. Creates an entry for each completed or terminated business process in the WF_INST_S table, and updates these tables: <ul style="list-style-type: none">◆ WORKFLOW_CONTEXT◆ DOCUMENT◆ DOCUMENT_EXTENSION◆ DATA_TABLE◆ CORRELATION_SET◆ WORKFLOW_LIFESPAN◆ DOCUMENT_LIFESPAN◆ WORKFLOW_DATA
Business usage	This service selects all complete and terminated business processes and flags them for archive or purge.
Usage example	Application runs a business process with the Index Business Process service information. BMPL can run manually or by schedule.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms.
Related services	Archive Business Process service, Purge service.
Application requirements	No
Initiates business processes?	No
Invocation	Runs by only the internal scheduler.
Business process context considerations	No

Returned status values	The following status values are returned after this service runs in a business process: <ul style="list-style-type: none"> ◆ 100s – Workflow engine error, for example, the service is deactivated ◆ 200s – System error, for example database error ◆ 300s – The service configuration error.
Restrictions	There can be only one configuration per server of this service.

Implementing the Index Business Process Service

To implement the Index Business Process service, complete the following tasks:

1. Create an Index Business Process service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Index Business Process service. For information, see *Configuring the Index Business Process Service* on page 707.
3. Use the Index Business Process service in a business process.

Configuring the Index Business Process Service

To configure the Index Business Process service, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Run As User	Enter the user ID to associate with this service when run.
Do not use schedule	If field is selected, it does not get scheduled to run.
Run service based on timer every	Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup.
Run service daily at	Valid values are the hour and minutes at which to run the service, daily. Indicate whether you want the service to run at startup.
Run service weekly on	Valid values are the day of the week, the hour, and the minutes at which to run the service. Indicate whether you want the service to run at startup.

Instant Messaging Adapter Suite

The Instant Messaging Adapter Suite enables you to use open standard instant messaging protocol, such as XMPP (which works with Jabber[®] instant messaging), to interact with Application. With the Instant Messaging Adapter Suite, Application can initiate instant messages to alert you of issues needing prompt attention. You can then act on the information, including initiating commands in Application and receiving the results of those commands within the context of the instant message (IM) session.

For the purposes of this documentation, the following terms are defined as follows:

IM user – User receiving the IM notification initiated by the Instant Messaging Adapter Suite

Application IM user – IM user account representing Application

The following table provides an overview of the Instant Messaging (IM) adapter suite:

System name	The IM Adapter Suite includes services having the following system names: <ul style="list-style-type: none">◆ IMBeginSession◆ IMBeginChat◆ IMSendMessage◆ IMReceiveMessage◆ IMGrammarLoader◆ IMCommandProcessor◆ IMEndChat◆ IMEndSession
Graphical Process Modeler (GPM) categories	All Services, Messaging > Instant Messaging Note: The IMGrammarLoader service is in the All Services category only.
Description	<p>The IM Adapter Suite enables you to carry on an IM session with Application as if the system were another IM user. You can configure multiple Application IM users.</p> <p>Using the IM Adapter Suite, you can send commands to Application, and receive the results of those commands in your IM display. You can pass files to and from Application using IM. To facilitate this, you define all of the commands you will use in a grammar file. The commands in the grammar file correlate to business processes that you create.</p> <p>You can configure the business process running the IM session to loop back after a command executes so that another command can be run. Recommended strategy is to define a command to end the session and make the command part of the IM business process. Application includes a default IM business process, grammar file, and business processes representing the commands in the grammar file. You can edit or copy any of these according to your needs.</p>
Business usage	Enables Application to alert a specified user of a certain event using instant messaging, and enables the alerted user to perform simple queries as to the health and status of the Application system through the IM session.

Usage example	<ol style="list-style-type: none"> 1 You configure the IM Adapter Suite services so that when Application receives an order from a top priority customer, the IM adapter suite will notify the appropriate user by initiating an instant message. 2 The system receives such an order. 3 The IM Adapter Suite notifies the user of the order.
Preconfigured?	<p>IM Adapter Suite services have the following service configurations provided with Application, however, some require further configuration to meet your needs, as indicated:</p> <ul style="list-style-type: none"> ◆ IMBeginChat_Instance (requires further configuration) ◆ IMGrammarLoader (requires further configuration) ◆ IMSendMessage_Instance ◆ IMReceiveMessage_Instance ◆ IMEndChat_Instance ◆ IMEndSession_Instance ◆ IMCommandProcessor_Instance ◆ IM_FILE_EXTRACT <p>For information about configuring the services, see <i>Configuring Instant Messaging Adapter Suite Services</i> on page 713.</p>
Requires third party files?	<p>Two .jar files are required:</p> <ul style="list-style-type: none"> ◆ jymsg available at http://jymsg9.sourceforge.net ◆ smack.jar, available from www.igniterealtime.org/downloads/ <p>Use the install3rdparty.sh or install3rdparty.cmd script to install the jar files. For information, see <i>Implementing the IM Adapter Suite</i> on page 711.</p> <p>To download the XMPP client libraries, go to www.igniterealtime.org/downloads/ and download the smack_3_0_4.zip file. For instructions, see <i>Implementing the IM Adapter Suite</i> on page 711.</p>
Platform availability	All supported Application platforms

Related services	<p>The IM Adapter Suite includes the following services:</p> <ul style="list-style-type: none"> ◆ IM Begin Session – Initiates the IM session as the Application IM user ID. ◆ IM Begin Chat – Calls the targeted IM user to have the IM session with. ◆ IM Send Message – Sends messages to the IM user. May send attachments to the IM user. ◆ IM Receive Message – Receives messages from the IM user. The system uses this service to pause and wait for an interactive message from the user. The receive message service waits for a message until the idle timeout expires. <p>Note: An IMReceiveMessage is not required to handle incoming attachments. Any time a user sends an attachment during an IM session, the IM adapter launches the IM_RECEIVE_FILE business process to handle the attachment.</p> <ul style="list-style-type: none"> ◆ IM Grammar Loader (File System adapter) – Loads the IM command syntax from the grammar file. The commands in the grammar file define the commands that the IM Command Processor will recognize. ◆ IM Command Processor – Receives input from the IM Receive Message service and interprets a received command according to the supplied grammar. When a command matches a grammar file entry, the Processor passes the appropriate business process name to invoke the process. <p>The value matched from command to grammar file becomes the parameter WFD_NAME in the business process, and a parsed version of the command becomes the primary document. Therefore, input to the IMCommandProcessor is the grammar XML as the primary document, and IMCommand as a process data parameter, and the output of the IMCommandProcessor is the parsed command as the primary document and the matching value as WFD_NAME.</p> <p>Note: Looping capability and business process launching and monitoring take place in the business process. The IM Command Processor only interprets the supplied command. The IMCommandProcessorLoop Business Process handles the looping mechanism and business process execution.</p> <ul style="list-style-type: none"> ◆ IM End Chat – Terminates the IM session visual display. ◆ IM End Session – Terminates the IM session communications.
Application requirements	<ul style="list-style-type: none"> ◆ Create a user account in the target instant messaging application on behalf of the Application system. This is the IM name you provide when configuring the IMBeginSession. ◆ Establish the user account in the target instant messaging application before running the SendMessage service.
Initiates business processes?	No
Invocation	<p>To enable command initiation using IM with the IM CommandProcessor, you must:</p> <ul style="list-style-type: none"> ◆ Write the corresponding business process for the commands ◆ Define the necessary commands in the IM grammar file
Business process context considerations	<p>IM Adapter Suite services rely upon session state stored in the business process. For that reason, the services must be used in conjunction with each other. IMBeginSession must always precede calls to IMBeginChat, and IMBeginChat must precede IMSendMessage and IMReceiveMessage. For an illustration, see <i>Business Process Example 1 – Simple</i> on page 715 and <i>Business Process Example 2 – Complex</i> on page 717.</p>

Returned status values	When errors occur, the business processes halt. The status report will specify, with as much detail as available, the failure point. This is vital because there are many process level state variables passed between services that are not manipulated by the user.
Restrictions	The system always initiates the IM session. Under no circumstances will Application respond to a session initiated by the IM user.
Persistence level	Recommended persistence level is None .
Testing considerations	You can create a business process similar to the first example to use for testing. See <i>Business Process Example 1 – Simple</i> on page 715.

Implementing the IM Adapter Suite

To implement the IM Adapter Suite, complete the following tasks:

1. With your IM provider, create a user ID representing Application.
2. Log in to your IM application, and add the ID to your messenger list
3. Log out and then log on as the new Application ID and add your personal user ID or the ID of another IM user to the messenger list of the Application user.
4. Log out of IM.
5. Obtain the third party XMPP client libraries by downloading the **smack_3_0_4.zip** file from www.igniterealtime.org/downloads/.
6. Unzip the file to your hard drive and note the location of the files.
7. Download the most recent version of the **ymsg_code_jars_v0_6.zip** file from <http://jymsg9.sourceforge.net>.
8. Unzip the file to your hard drive and note the location of the files.
9. Shut down the application.
10. Unload the following jar files you unzipped in steps 6 and 8:

- ◆ smack.jar
- ◆ ymsg_network_v0_6.jar
- ◆ ymsg_support_v0_6.jar

To unload the files, use `install3rdParty.sh` (or `install3rdParty.cmd` for Windows). The following examples show the commands; you must change the versions and user directories according to your needs:

- ◆ `./install3rdParty.sh ymsg_network 0_6 -j /unzip directory/ymsg_network_v0_6.jar`
- ◆ `./install3rdParty.sh ymsg_support 0_6 -j /unzip directory`
- ◆ `./install3rdParty.sh smack 3_0 -j /unzip directory/smack.jar`

11. Restart the application.
12. Do you need the adapter to accept to commands from the IM user during sessions?

- ◆ If Yes, define the commands in the grammar file. For information, see *Default Grammar File* on page 712.
 - ◆ If No, skip to step 15.
13. In Application, edit the IMGrammarLoader configuration of the File System adapter. In the **attachFile** field, enter the location and name of the grammar file to use. This is the file you modified in step 12.
 14. In the GPM, configure the Command Processor service. For information, see *Configuring the IM Command Processor Service* on page 715.
 15. In Application, create a configuration of the IM Begin Session service. Enter the IM user ID and password to represent Application in IM sessions. For information, see *Configuring the IM Begin Session Service* on page 713.
 16. In Application or in the GPM, edit the IM Begin Chat service configuration. Enter the IM user ID to connect with during the session (this is your ID or the IM user ID of another user that Application will notify). You can enter multiple IDs by separating them with commas. For example, mychatid, yourchatid, thirdchatid. For information, see *Configuring the IM Begin Chat Service* on page 714.
 17. Use the IM services in a business process.

Default Grammar File

The provided grammar file for the IM adapter is an XML-format file that describes the behavior of the IM Command Processor service. The file, `grammar.xml`, is located in the `data/xmpp` directory of your Application installation directory.

You can copy and modify this file as necessary in order to define the commands to be used. This enables the Command Processor to match the correct business processes to commands received in an IM session. If you use a different grammar file or change the location of the file, you must modify the IMCommandProcessor business process to reflect the filename or path change.

The contents of the default grammar file are as follow:

```
<grammar>
<pattern value="get [a-z]* log" result="IM_GET_LOG"/>
<pattern value="show [a-z]*" result="IM_SHOW_COMMAND"/>
<pattern value="bye" result="IM_EXIT"/>
<pattern value="default" result="IM_DEFAULT"/>
</grammar>
```

The grammar file contains two tag types:

Root – This is always ‘grammar.’

Pattern – A repeatable tag comprised of two required attributes, *value* and *result*. This pairing describes the action taken when the system receives a given command.

The content of ‘value’ is a Perl5 type regular expression that will match on command phrases. The content of ‘result’ is the business process to execute when the value is matched. In the following example, this is illustrated in the first pattern:

```
<pattern value="get [a-z]* log" result="IM_GET_LOG"/>
```

The match will execute the IM_GET_LOG business process when it receives a command of:

```
get<space><any combination of letters><space>log
```

Note: The last pattern in the grammar file, with a value of default, is required in every grammar file. This is the pattern the Command Processor uses when it finds no matching command.

Example Commands

The following table provides examples of commands that will execute the IM_GET_LOG business process and commands that will not:

Valid command	Invalid command
get noapp log	get the big log
get all log	get all logs
get myspecial log	get log

Configuring Instant Messaging Adapter Suite Services

Several of the IM Adapter Suite services require configuration.

Configuring the IM Begin Session Service

To configure the IM Begin Session service, you must specify field settings in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they display in the list. Select a group from the list.
Protocol	IM provider to use for chat sessions. Select XMPP or Yahoo . Default is XMPP . Required. Note: Yahoo IM is not supported at this time.

Field	Description
Username	IM user ID name to represent Application use during IM sessions. Required.
Password	IM password for the Application user ID. Optional.
Idle Timeout	Length of time in milliseconds to wait for a response from the IM user before halting the process and ending the chat session. Optional.
Session Timeout	Length of time in minutes to allow for the entire session to run before halting the process and ending the chat session. Optional.
XMPP Server Name	Name of the server of the chat service you are using. Obtain this name from your IM provider. Required.
XMPP Server Port	Port number for IM communications. Obtain the port number from your IM provider. Required. Note: The default port for XMPP with Jabber is 5222.
XMPP Service Name	XMPP service name as defined in the XMPP server if different from the XMPP Server Name value. If omitted, the adapter uses the XMPP Server Name. Optional.

Configuring the IM Begin Chat Service

To configure the IM Begin Chat service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration. Required.
IMChatTo	Name of the IM user for Application to have a chat session with, or a comma delimited list of IM user IDs, for an IM conference. Required.

Configuring the IM Send Message Service

To configure the IM Send Message service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration. Required.
IMBodyAsAttachment	Valid values are True and False . When the value is set to True, the system sends the primary document body as an attachment during a SendMessage invocation. When the value is set to False, the system chunks the primary document into multiple messages and sends them to the user. The chunk size is determined by the maximum message size for the IM provider. Use this field only when IMUseBody is set to True. Optional.
IMMessage	Message to display on the screen of the IM user. Use this field if you want to specify exactly what the notification will say when Application initiates a chat session. Optional.
IMUseBody	Valid values are True and False . When the value is set to True, the system sends the body of the primary document to the user according to the mode specified in the IMBodyAsAttachment field. Optional.

Configuring the IM Grammar Loader Service

To configure the IM Grammar Loader service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration. Required.
attachFile	Name of the grammar file for the adapter to reference during the session. Required.

Configuring the IM Command Processor Service

To configure the IM Command Processor service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration. Required.
IMCommand	Specifies the command for the command processor to match against. Used in conjunction with <code>IMReceiveMessage</code> , this parameter automatically populates according to the message received from the user. Because this is a system-populated value, do not enter a value. Entering a value will force the service to use your supplied value regardless of the actual command received from the IM user.

Instant Messaging Adapter Suite Business Processes

The following table describes the predefined business processes associated with the IM Adapter Suite:

BPML Name	Usage Description
IM_DEFAULT.bpml	In the event that the adapter receives a command for which there is no match in the grammar file, this is the default process that runs, to indicate that the system does not understand the command.
IM_EXIT.bpml	Closes the chat session when the IM user enters an exit command.
IM_GET_LOG.bpml	Example business process that returns a Application log file.
IM_SHOW_COMMAND.bpml	Lists all commands in the grammar file.
IMCommandProcessLoop.bpml	Example process you can configure to run the looping process for proper handling of user-supplied commands within a chat session.

Example Business Process Models

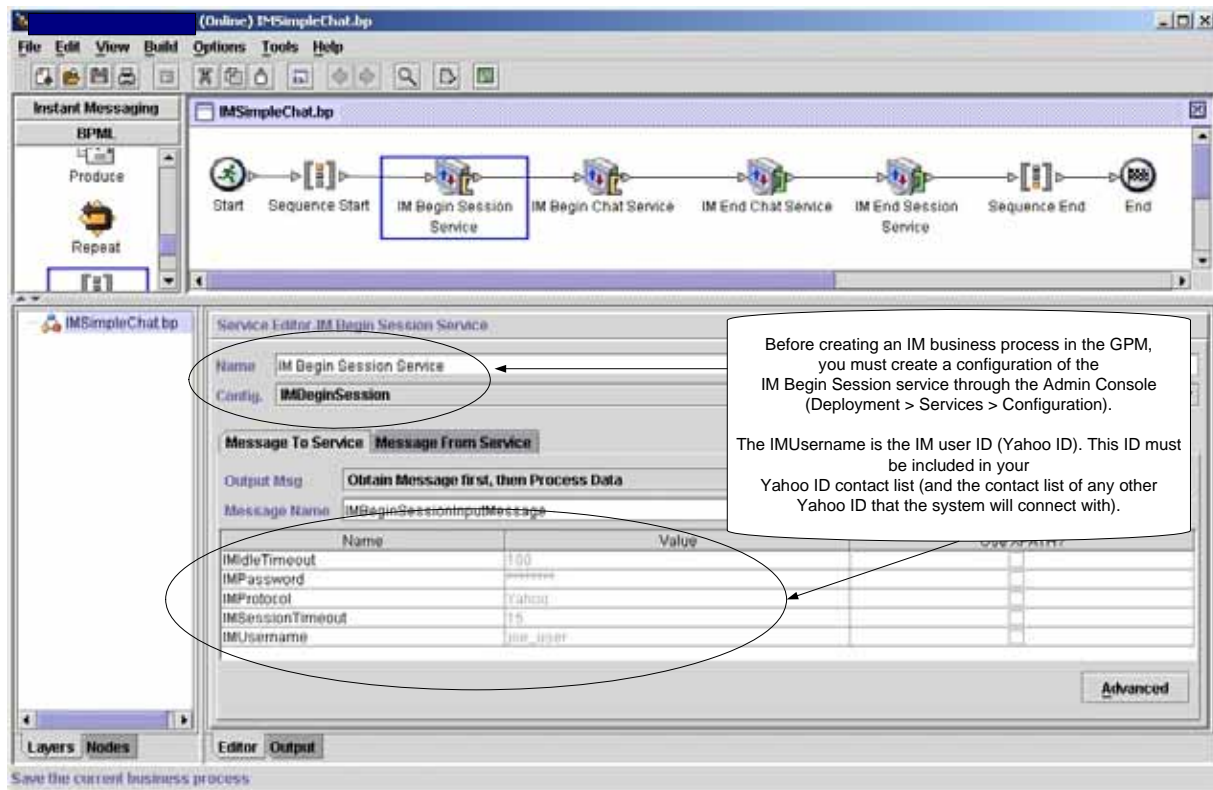
This section contains two examples, a simple connect-chat-disconnect business process that you can build and use to test your IM connections, and a more complex example.

Business Process Example 1 – Simple

Refer to the figure following the steps as you read the process steps. This example business process progresses through IM Adapter Suite services as follows:

1. IM Begin Session service – This service contains the Application IM user ID and password used to log in to IM, and sets basic session parameters. The system logs into IM using the specified ID and password.
2. IM Begin Chat service – This service contains the IM user ID (or comma-delimited list of IDs) the system communicates to when initiating an IM session. The system starts a chat session with the IDs specified.
3. IM End Chat service – When the IM user closes the chat session, this service ends the chat session.
4. IM End Session service – When the chat session has ended, this service signs the Application user ID out of the IM provider application.

The following figure shows the business process in the Graphical Process Modeler:



The BPML for the business process would be similar to the following example:

```
<process name="default">
  <sequence>
    <operation name="IM Begin Session Service">
      <participant name="IMBeginSession"/>
      <output message="IMBeginSessionInputMessage">
        <assign to="." from="*"></assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="IM Begin Chat Service">
    <participant name="IMBeginChat_Instance"/>
    <output message="IMBeginChatInputMessage">
      <assign to="IMChatTo">myuserid, TomR_Billing, WarehouseUser1</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="IM End Chat Service">
    <participant name="IMEndChat_Instance"/>
    <output message="IMEndChatInputMessage">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="IM End Session Service">
    <participant name="IMEndSession_Instance"/>
    <output message="IMEndSessionInputMessage">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

Business Process Example 2 – Complex

This example corresponds with the sample business process `IMCommandProcessLoop.bp` provided with your installation of Application. To use the provided process model and this example, use a configured instance of the IM Begin Session service to call the process.

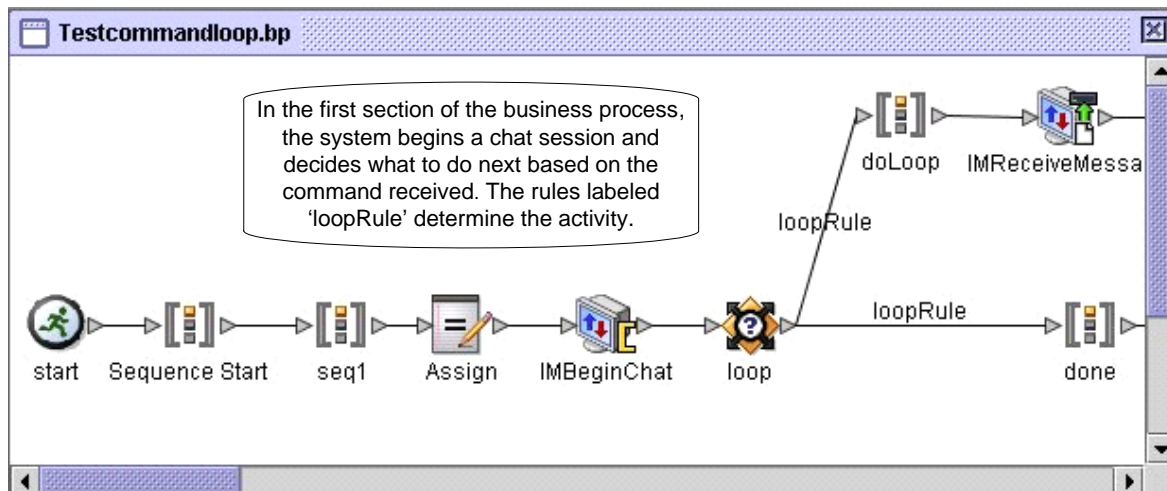
Refer to the figures following the steps as you read the process steps. This example business process progresses through IM Adapter Suite services as follows:

1. The session begins and you see a message in your IM window.
2. You type in a command, which the system evaluates at the decision point in the business process model according to the rule, illustrated as the `loopRule` (your process model may use different labels).
3. The the grammar file is loaded and the IM Command Processor service attempts to associate the command with a match in the grammar file:

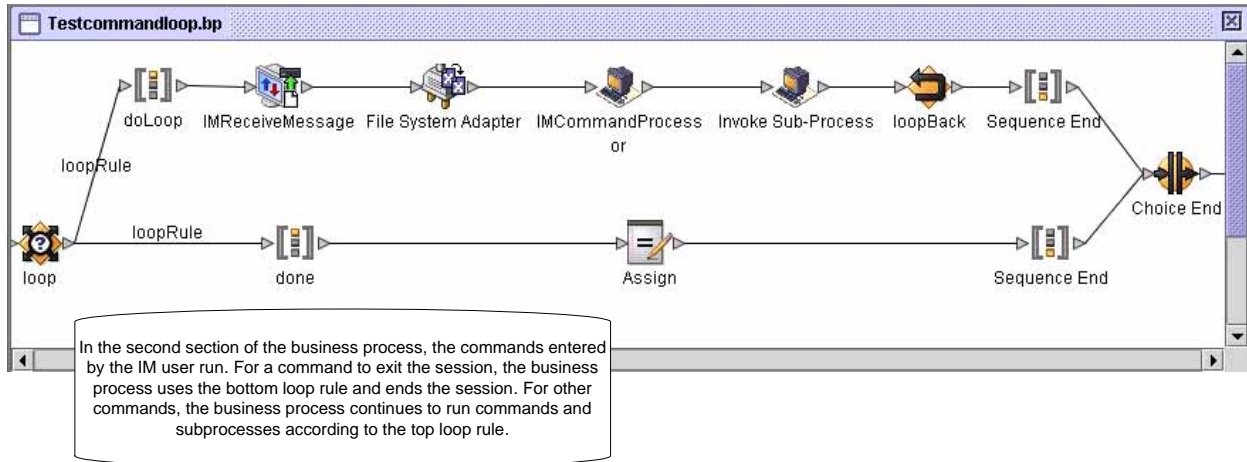
- ◆ If the IM Command Processor service finds a match, it starts the corresponding business process. You receive any output of the business process in your IM window.
 - ◆ If the IM Command Processor service does not find a match, it sends you an IM message, saying that the match was not found.
4. The next step depends on the command you sent in step 2:
- ◆ If you sent an end session command, the IM Command Processor ends the session and exits the business process.
 - ◆ If you sent a command other than to end the session, the IM Command Processor restarts the command loop and the business process waits for another command.

The following figure shows a similar business process model in the GPM. Because of the length of the business process, the figure is broken down into three separate sections here for ease of viewing:

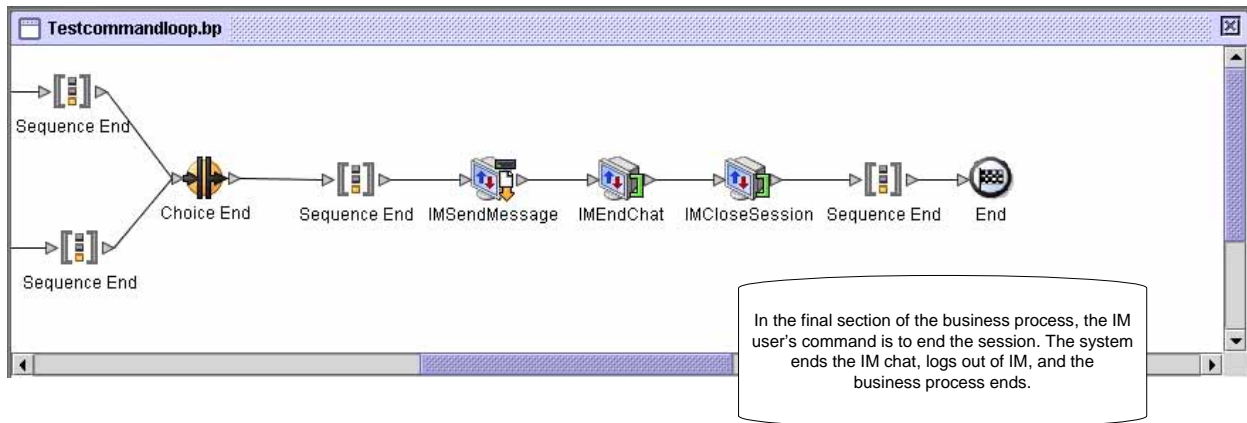
Section 1



Section 2



Section 3



Invoke Sub-Process Service

Note: This service is also referred to as the Invoke Business Process service and the Invoke service.

The following table provides an overview of the Invoke Sub-Process service:

System name	Invoke Sub-Process Service
Graphical Process Modeler (GPM) category	All Services, Sync Mode, Transactional Mode
Description	Starts a subprocess.
Business usage	The Invoke Sub-Process service provides a way to reuse the same business process in multiple business processes by enabling you to invoke the subprocess from within a parent process.
Usage example	Runs a standard deenvolving (for example, EDI or SOAP) business process to extract business payload from a document bundle for transmission.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	Yes. Each call to the Invoke Sub-Process service starts one subprocess (child). The Invoke Sub-Process service can be called multiple times from a business process. Subprocesses can be started in asynchronous or synchronous mode. Subprocesses can optionally be run as separate processes, or inline as part of the parent business process.
Invocation	If a child process is disabled, a service is disabled, or there is a license error causing the subprocess to halt, the parent will not continue. Additionally, if the subprocess is manually stopped, or stopped using a system shutdown, the parent will not continue. Note: You can fix the failures to continue the parent by resuming the subprocess.
Business process context considerations	The Invoke Sub-Process service adds the business process instance ID to the INVOKE_ID_LIST object stored in the business process context. The Invoke Sub-Process service also places the business process instance ID of the generated sub-process into the IWF_Id name/value pair and stores it in the business process context. The tracking feature uses this value to be able to show information about subprocesses. This value is then deleted from the business process context before any services that follow are run.

Returned status values	<ul style="list-style-type: none"> ◆ Success – If the subprocess was started correctly (async, inline modes) or if the child executed successfully (sync or inline modes). ◆ Error – If the subprocess was not started correctly (async) or if the child did not execute successfully (all modes). <p>When running in embedded mode:</p> <ul style="list-style-type: none"> ◆ Error – From the error step in the subprocess, if the subprocess ended with an error. ◆ Success – if the subprocess completed.
Restrictions	<p>Restrictions by mode are:</p> <ul style="list-style-type: none"> ◆ Synchronous <p>When the Invoke Sub-Process service is set to synchronous mode, the parent business process suspends processing until it receives data from the child business process. In synchronous mode, the parent is notified when the child encounters errors.</p> ◆ Asynchronous <p>When the Invoke Sub-Process service is set to asynchronous mode, the parent and child process simultaneously and independently of each other. Therefore, the parent does not receive notification when the child encounters errors.</p> ◆ Inline <p>When the Invoke Sub-Process service is set to run a subprocess inline, the subprocess runs as part of the parent process, sharing the same process data.</p> ◆ Embedded <p>Use to run a subprocess with no persistence.</p> <p>Note: A subprocess run inline will be displayed in tracking just like other steps in the parent process.</p>
Persistence level	Full (except in Embedded mode, which runs the subprocess with no persistence)
Testing considerations	None

How the Invoke Sub-Process Service Works

When the Invoke Sub-Process service is set to synchronous mode, the parent suspends processing until it receives data from the child. In synchronous mode, the parent is notified when the child encounters errors.

When the Invoke Sub-Process service is set to asynchronous mode, the parent and child process data simultaneously and independently of each other. Therefore, the parent does not receive notification when the child encounters errors.

When the Invoke Sub-Process service is set to run a subprocess inline, the subprocess runs as part of the parent process, sharing the same process data.

When the Invoke Sub-Process service is set to run in embedded mode, the subprocess runs with no persistence, meaning that no record of the process is recorded in Application and no tracking is done.

Performance Tips

By default, when you use the Invoke Sub-Process service in a business process, all process data passes from the parent process to its subprocess.

However, if you are using the Invoke Sub-Process service in sync mode, a special tag called 'message_to_child/message_to_parent' enables you to pass along only the 'message_to_child/message_to_parent' node in the process data of the parent process or subprocess. Using this tag can provide significant performance improvement.

Before invoking a subprocess, create a special tag called 'message_to_child' in the parent process, and append all of the data needed in the subprocess under this node. The Invoke Sub-Process service will pass only this node to the subprocess. The following example shows how the BPML might look for this:

```
<process name="MessageToChildInvokeDoc">
  <sequence name="simple">
    <assign to="Msg1" from="'Hello' " append="true"/>
    <assign to="Msg2" from="Msg1" append="true"/>
    <assign to="Msg3" from="Msg1"/>
    <assign to="OtherMsg4" from="//Msg2" append="true"/>
    <assign to="OtherMsg5/OtherMsg6" from="//Msg2/Msg1/text()" append="true"/>
    <assign to="message_to_child" from="//PrimaryDocument | // OtherMsg5 "
append="true"/>
  <operation>
    <participant name="InvokeBusinessProcessService"/>
    <output message="Xout">
      <assign to="." from="*"></assign>
      <assign to="WFD_NAME">MessageToParentNode</assign>
      <assign to="INVOKE_MODE">SYNC</assign>
    </output>
    <input message="Xin" >
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
</process>
```

Or, you can specify the return node from the input message in the Invoke Sub-Process service. That is, only the ChildDoc node will be returned from the subprocess. In this case, you do not need to specify the 'message_to_parent' node in the subprocess. This is shown in the following example:

```
<operation>
  <participant name="InvokeBusinessProcessService"/>
  <output message="Xout">
    <assign to="." from="*"></assign>
    <assign to="WFD_NAME">MessageToParentNode</assign>
    <assign to="INVOKE_MODE">SYNC</assign>
  </output>
  <input message="Xin" >
    <assign to="test1" from="//ChildDoc"></assign>
    --- this 'ChildDoc' node is the sub node of the process data in its sub process
    MessageToParentNode-----
  </input>
</operation>
```

If you are not specifying the return node in input message in the Invoke Sub-Process service, you can, in the subprocess, create a tag 'message_to_parent' and append all of the data needed from the subprocess to the parent process under this node. Only data under this node will be returned to the parent process. The default is passing all process data to the parent process.

Example

```
<process name="MessageToParentNode">
  <sequence name="simple">
    <assign to="ChildMsg1" from="'Hello World'" append="true"/>
    <assign to="ChildMsg2" from="ChildMsg1" append="true"/>
    <assign to="ChildMsg3" from="ChildMsg1"/>
    <assign to="ChildDoc" from="PrimaryDocument/@SCIOBJECTID" append="true"/>
    <operation name="Set Document">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">process_data_to_document</assign>
        <assign to="root_element">DocumentSub</assign>
        <assign to="xPath">/ProcessData</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="xmldoc" from="PrimaryDocument" append="true"/>
      </input>
    </operation>
    <assign to="message_to_parent" from="//PrimaryDocument | //ChildMsg3" />
  </sequence>
</process>
```

Implementing the Invoke Sub-Process Service

To implement the Invoke Sub-Process service for use in a business process, complete the following tasks:

1. Create a configuration of the Invoke Sub-Process service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Invoke Sub-Process Service*.
2. Specify field settings for the service configuration in the Application Admin Console and in the GPM as necessary. For information, see *Configuring the Invoke Sub-Process Service*.
3. Use the Invoke Sub-Process service in a business process.

Configuring the Invoke Sub-Process Service

To configure the Invoke Sub-Process service, you must specify settings for the following fields in the GPM.

Note: Any field values passed from a prior service can override any of configured fields for this service.

Field	Description
Config	Name of the service configuration. Required.
WFD_NAME	Business process used in the service configuration. Valid value is the name of any business process checked into the system. Required.

Field	Description
INVOKE_MODE	<p>Mode in which to run the subprocess. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ async = asynchronous (default) ◆ sync = synchronous ◆ inline = In-line ◆ embedded = starts business process in SYNC mode with Enable Transaction on, and runs the subprocess in the same transaction as the Invoke Sub-Process service. The subprocess is run with no persistence. <p>The following optional parameters can be used with embedded mode:</p> <ul style="list-style-type: none"> ◆ START_WITH_DOC – Use if the subprocess needs to be started with a document. The process checks the DOC_URL parameter for a filename (see following parameter) first. If the DOC_URL parameter is set to NONE, then the primary document is passed to the subprocess. This parameter is required if the subprocess needs a document from the parent business process. ◆ DOC_URL – Name of the document from which the subprocess will retrieve data. Use this parameter with the START_WITH_DOC parameter if the parent primary document is not the document needed to start the subprocess. ◆ DOC_ENCODING – Use if the document that is used to start the subprocess require different encoding from the default encoding. ◆ USER_NAME – Use if the subprocess requires a different user name to start (permission to start). ◆ PASS_DOC – Use if the subprocess needs to pass its primary document to back to the Invoke Sub-Process service.
PARAM_LIST	<p>List of business process parameters to override. Valid values are:</p> <ul style="list-style-type: none"> ◆ Name1=value1 ◆ &name2=value2 ◆ &name3=value3 <p>Optional.</p>

Field	Description
NOTIFY_PARENT_ON_ERROR	<p>Errors the subprocess reports to the parent business process (synchronous mode only). Valid values are:</p> <ul style="list-style-type: none"> ◆ NOTIFY_PARENT_ON_ERROR_SERVICE_ERRORS_ONLY = SERVICE - Notify parent of the error only if a service-generated error occurs in the subprocess. ◆ NOTIFY_PARENT_ON_ERROR_ALL = ALL - Notify parent of the error if any type of error occurs in the subprocess. ◆ NOTIFY_PARENT_ON_ERROR_NONE =NONE - This option does not notify the parent of the error from the subprocess. The subprocess completes with status error. ◆ NOTIFY_PARENT_ON_ERROR_SYSTEM_ERRORS_ONLY = SYSTEM - Notify parent of the error only if a system-generated error occurs in the subprocess. <p>Default is NOTIFY_PARENT_ON_ERROR_ALL = ALL. Optional.</p>
PASS_STATUS_REPORT	<p>Pass status report to parent. Valid values are:</p> <ul style="list-style-type: none"> ◆ ON_SUCCESS ◆ ON_ERROR ◆ ALWAYS ◆ NEVER <p>Default is ON_ERROR. Required.</p>
COPY_SERVICE_PARMS	<p>Invoke service parameters passed to the subprocess. Valid values are True and False. Default is True. Required.</p>

Parameters That Must Be Added in BPML

The following additional parameter is available for use with the Invoke Sub-Process service, but can only be added by editing your business process manually. This parameter is not available through the Admin console or the GPM:

Parameter	Description
SAME_TRANSACTION	<p>Use this parameter to set a sub-business process to allow the sub-business process to error if the Invoke service has an error status. SAME_TRANSACTION should be manually set to false. The default is true.</p> <p>The 'same-transaction'=false.sub business process will be started, and only the first step (step id 0) along with the data passed from invokeservice will be saved. The sub-business process may have an active state or error state.</p> <p>When business process definition has the following setting, SAME_TRANSACTION will not be valid from the Invoke service. The system will internally set it to false:</p> <ul style="list-style-type: none">◆ startMode =Sync◆ Transaction =TRUE◆ persistence_level=PERSISTENCE_WF_NONE,◆ persistence_level=PERSISTENCE_ERROR_ONLY

Business Process Example

The following BPML example illustrates using the SAME_TRANSACTION parameter:

```
<operation>
  <participant name="InvokeBusinessProcessService"/>
  <output message="Xout" >
    <assign to="INVOKE_MODE">SYNC</assign>
    <assign to="WFD_NAME">sleepy.bpml</assign>
    <assign to="SAME_TRANSACTION">>false</assign>
  </output>
  <input message="Xin" >
    <assign to="." from="*"></assign>
  </input>
</operation>
```

Output from Service to Business Process

The following table describes the output from the Invoke Sub-Process service to the business process:

Parameter	Description
INVOKE_ID_LIST	A generated list of business process IDs for the processes running.
WFD_VERSION	Version of the business process captured by the system at run time. This is a system assigned parameter and is not configurable.

Business Process Example

The following example illustrates how the Invoke Sub-Process service could be used in a business process with Async mode:

```
<operation>
  <participant name="InvokeBusinessProcessService" />
    <output message="Xout">
      <assign to="INVOKE_MODE">ASYNC</assign>
      <assign to="WFD_NAME">mybusprocess</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin" >
      <assign to="." from="*"></assign>
    </input>
</operation>
```

Running a Business Process with No Persistence

To enable a business process to run with no persistence, you must start it as a subprocess (child) of another business process (parent). Include the Invoke Sub-Process service in the parent business process, and set the service to use Embedded mode. Embedded mode starts the business process in SYNC mode with Enable Transaction on, and runs the subprocess in the same transaction as the Invoke Sub-Process service. The subprocess is run with no persistence.

When checking in the parent process to Application, on the Process levels page, make sure that:

- Enable Async Start Mode Select is not selected (so that the business process starts in SYNC mode)

- Enable Transaction is selected

The following example illustrates using the Embedded mode:

```
<operation>
<participant name="InvokeBusinessProcessService" />
<output message="Xout">
<assign to="INVOKE_MODE">EMBEDDED</assign>
<assign to="WFD_NAME">mybusprocess</assign>
<assign to="." from="*"></assign>
</output>
<input message="Xin" >
<assign to="." from="*"></assign>
</input>
</operation>
```

iWay Adapter

The following table provides an overview of the iWay adapter:

System name	iWayadapter
Graphical Process Modeler (GPM) categories	All Services, Applications > ERP
Description	Enables Application to communicate with over 100 adapters that iWay communicates with through the iWay XML Transformation Engine (iXTE) server. Adapter types supported by the iWay adapter include application systems adapters (ERP and CRM) and data adapters (relational data and connector technologies).
Business usage	Sends documents from backend systems through Application to iWay.
Usage example	You are sending data to a PeopleSoft® application through iWay. The data is sent from your system to Application, which translates the data into an XML document. The business process runs the iWay adapter, which takes the data, envelopes it, and sends it to iWay.
Preconfigured?	No
Requires third party files?	Yes. The iWay Application Explorer version 5.2.1 or higher is needed to create XML schemas that can be used in a Application translation map.
Platform availability	All supported Application platforms
Related services	XML Translation service
Application requirements	Application supports iWay version 5.2.1 and higher. To use the iWay adapter, you must have: <ul style="list-style-type: none">◆ iWay software installed and configured◆ iWay documentation◆ The appropriate adapter documentation
Initiates business processes?	Yes
Invocation	Runs as part of a business process.
Business process context considerations	Resultant workflow context contains response from iWay adapter. It does not contain the original request. In bootstrap mode, a workflow context is created by the iWay adapter.

Returned status values	Success, failure A status report is generated on failure.
Restrictions	None
Persistence level	System default (Full)
Testing considerations	To test outbound, an iWay instance must be running and a listener configured on the iWay server. To test inbound, the iWay adapter agent must be installed on the iWay server.

Requirements

Application supports iWay version 5.2.1 and higher. To use the iWay adapter, you must have:

- iWay software installed and configured
- iWay documentation
- The appropriate adapter documentation

For information about iWay software, see www.iWaysoftware.com.

Before Using the iWay Adapter

Before you can use the iWay adapter, you must create XML schemas and translation maps to generate XML documents that can be sent to iWay. For each iWay configuration you have, use the iWay Application Explorer to create two XML schemas: one for Request documents and one for Response documents. For example, you may have separate iWay configurations for PeopleSoft, UCCnet, and SAP.

1. Determine how many iWay configurations you will integrate with Application by using the iWay adapter.
2. Use the iWay Application Explorer to create Request and Response XML schemas for each.
3. Check in the XML schemas to Application.
4. Using the Application Map Editor, create maps for each schema to translate from your document format to the format required for iWay.
5. Test the maps to verify that no translation errors exist.

For more information about the iWay Application Explorer, see your iWay documentation or the iWay website: <http://www.iWaysoftware.com>.

How the iWay Adapter Works

The following steps summarize how the iWay adapter works in an Application business process:

1. The business process passes XML data containing the executable command to the iWay adapter.
2. The iWay adapter envelopes the incoming data into an iWay RequestXML document, and sends this request to the iWay Software iXTE server.
3. The iWay adapter waits for a response from the iWay Software iXTE server.
4. The iWay Software iXTE server receives the RequestXML document and passes it on to the appropriate iWay Software Intelligent adapter using a preconfigured datasource.
5. The iWay Software Intelligent adapter executes the request against the back-end system and returns an answer set to the iWay Software iXTE server.
6. The iWay Software iXTE server passes the results back to the Application iWay adapter in the form of a ResponseXML document.
7. The iWay adapter parses the response (ResponseXML) and places the data into the output business process context. The status of the output business process context is SUCCESS and the primary document contains the result set.

Note: Not all commands produce a result set.

Note the following exceptions:

- ◆ If the XML data produces an empty result set (as with delete or update requests), the status of the output business process context is SUCCESS and the primary document contains the empty result set.
 - ◆ If the returned ResponseXML document contains an error, the ResponseXML document is parsed and the data is placed into the output business process context. The status of the output business process context is ERROR and the primary document contains the RequestXML document data.
 - ◆ If the returned ResponseXML document is not received within the specified time (default 30 seconds), the status of the output business process context is ERROR and the primary document contains the Request XML document data.
8. The business process continues with the next activity.

Implementing the iWay Adapter

To implement the iWay adapter, complete the following tasks:

1. Activate your license for the iWay adapter. See *An Overview of Implementing Services*.
2. Create an iWay adapter configuration. See *Creating a Service Configuration*.
3. Configure the iWay adapter. See *Configuring the iWay Adapter* on page 731.
4. Create and enable a business process that includes the iWay adapter configuration.

Note: The business process must be set up to move data between Application and one or more of the systems supported by the iWay adapter.

5. Test the business process and the adapter.
6. Run the business process.

Configuring the iWay Adapter

To configure the iWay adapter, you must specify field settings in Application:

Note: The names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: See <i>Using Service Groups</i> .
iWay hostname (HostName)	Host name or IP address of the iWay iXTE server. Required.
iWay port number (PortNumber)	Port number of the iWay iXTE server listener. Required.
Inbound Service Listen Port (listenPort)	Enter the port number for the iWay adapter to listen on. Required.
Inbound Service Bind Address (listenHostname)	Host name and port number for the iWay adapter to listen on. Optional. Note: This parameter provides multi-homed server support and allows tighter security. By default, the iWay adapter will bind to all available homes (network interfaces) on the iWay iXTE server and listen on all of them. If a host is entered for this parameter, the iWay adapter will only listen on that host. If left blank, the adapter will use the default behavior and listen on all available homes.
Bootstrap Process (bpID)	If using the adapter to initiate a business process, select the name of the business process from the list.
Document Storage Type (docStorageType)	Defines how the document will be stored in the system. Required Valid values: <ul style="list-style-type: none">◆ System Default◆ Database◆ File System Note: See <i>Selecting a Document Storage Method for Bootstrap Adapters</i> .

Deploying the GIS iWay Agent

To deploy the GIS iWay Agent on the iWay iXTE server:

1. Copy the *GIS_Installation/client/iWay/GISiWayAgent.jar* file to the system that contains the iXTE server.
2. From the iXTE console, select **Configuration > Register Libraries**
3. Add the GISiWayAgent.jar file from this interface.
4. Select **Configuration > Defines > Agent**
5. In the New Agent dialog box, enter the following:
 - ◆ For **Alias**, enter **GISAgent**
 - ◆ For **Procedure**, enter `com.sterlingcommerce.woodstock.services.iWay.agent.GISRequestAgent (gisservname,iWay adapter Listen port,GIS User,GIS Password)`

For example, assume Application is running on a machine named **fred** and the iWay adapter is configured to listen on port **50000** and you have a user called **joe** with a password of **wilma**. The resulting procedure setting would be:

`com.sterlingcommerce.woodstock.services.iWay.agent.GISRequestAgent (fred,50000,joe,wilma)`
 - ◆ For **Comment**, enter **GIS Request Agent**
6. Configure your listener to use this agent:
 - a. Go to **Configuration/Listeners** and select the listener to configure.
 - b. In the upper right hand corner, click **Agents**.
 - c. Select **Add Agents > GISAgent**.
 - d. Click **Add**.

Anything received by this listener will now be forwarded to Application.

Note: The Application iWay Agent only works with XML payloads.

Java Database Connectivity (JDBC) Adapter

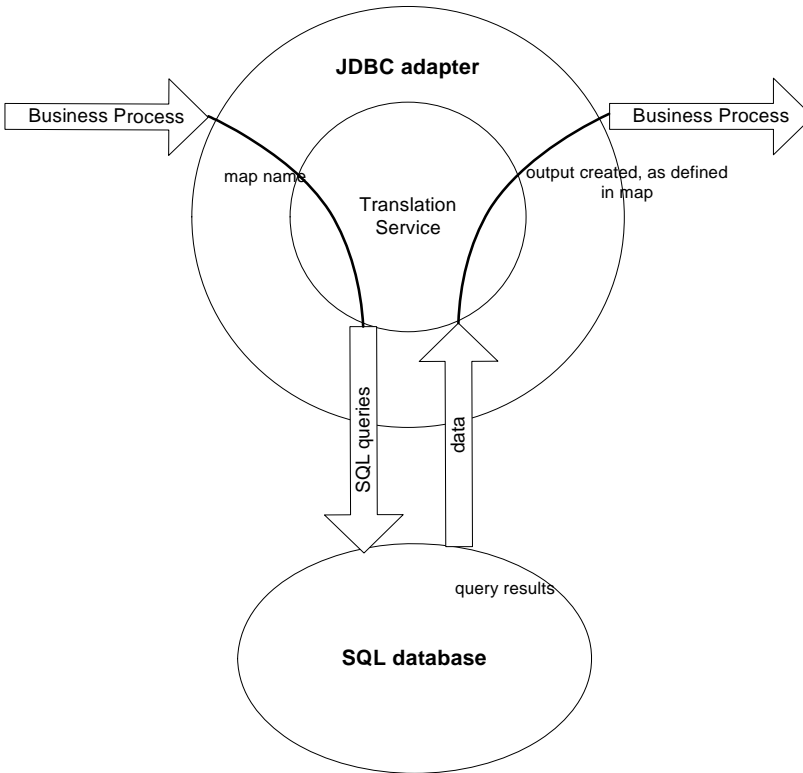
The Java Database Connectivity (JDBC) adapter enables the Translation service to communicate with JDBC-compliant databases. The adapter enables you to update or retrieve data from a JDBC-compliant database as part of a business process within the Application. The following table provides an overview of the JDBC adapter:

System name	JDBCAdapterType
Graphical Process Modeler (GPM) category	All Services
Description	Queries data from a remote database.
Business usage	Used to query or update data from a remote JDBC database by calling the Translator service.
Usage example	To perform any type of database query and return the results.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	To use the JDBC adapter, the <code>jdbc_customer.properties</code> file must be edited to reference a valid data source and database. Also, the map that is going to be used by the translator must be checked in to the Application.
Initiates business processes?	If configured to start a new business process, the results returned from the database query will be used to start a new business process.
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	<ul style="list-style-type: none">◆ Success – JDBC adapter finished successfully.◆ Warning – JDBC adapter finished but with warnings.◆ Error – JDBC adapter finished, but with errors.
Restrictions	None

How the JDBC Adapter Works

The queries you define in your map determine the data that to be retrieved or updated. You can submit any query written in SQL, stored procedures, or stored functions to a database. The business process you create then determines how the data is used.

The following figure shows how the JDBC adapter communicates with an SQL database within a business process:



JDBC Adapter Business Process Usage

The JDBC adapter can start a business process, or it can be used in the middle or at the end of a business process. The following steps summarize how the JDBC adapter might be used in a business process:

1. The JDBC adapter receives a map name from the business process.
2. The adapter starts the Translation service and passes the name of the map to the translator.
3. Using the map, the translator creates the SQL commands and sends them to an SQL database.
4. The translator receives the results from an SQL database query, creates the output as defined in the map, and passes those results back to the JDBC adapter.
5. The business process advances to the next step.

Example

For example, you have customer information stored in a confidential database. Your Sales department does not have access to the database. You can use the JDBC adapter to provide the Sales department with access to customer information in the database and then write the information to disk using the File System adapter.

The following steps summarize the JDBC adapter data flow for this example:

1. The adapter receives the name of the map from the business process.
2. The adapter starts the Translation service and passes it the map name.
3. The Translation service executes the map and generates SQL queries to submit to the database.
4. The Translation service submits the SQL queries to the database.
5. The Translation service receives an SQL response from the database and performs another translation to establish that the response is in a format that the adapter can process.
6. The Translation service passes the translated response to the adapter.
7. The adapter sends the customer information to the next step in the business process, the File System adapter.
8. The File System adapter writes the retrieved customer information to disk for the Sales department.
9. The Application performs the next activity in the business process.

Implementing the JDBC Adapter

To implement the JDBC adapter, complete the following tasks:

1. Create a JDBC adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the JDBC adapter. For information, see *Configuring the JDBC Adapter* on page 735.
3. Install the appropriate database drivers, if necessary, on the same computer where the Map Editor was installed.
4. Create an ODBC data source. For more information about ODBC, access www.msdn.microsoft.com and locate the ODBC Programming Reference documentation.
5. Set up a connection to an external database. For information, see *Setting Up a Connection to an External Database* on page 738.
6. Create either an input or output map using the Map Editor.
7. Check in the map for versioning control.
8. Use the JDBC adapter in a business process.

Configuring the JDBC Adapter

The Application Configuration

The following table describes the fields used to configure the JDBC adapter in the Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Start a new business process (StartNewWorkFlow)	<p>Whether to start a new business process.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> ◆ Yes: This JDBC adapter starts a new business process. ◆ No: This JDBC adapter does not start a new business process.
Business Process	<p>Select the business process that this adapter will start.</p> <p>Only applicable when Start a new business process is set to Yes.</p>
<ul style="list-style-type: none"> ◆ EDI Output Tag Delimiter (edi_output_tag_delimiter) ◆ EDI Output Segment Terminator (edi_output_segment_delimiter) ◆ EDI Output Data Element Separator (edi_output_element_delimiter) ◆ EDI Output Component Element Separator (edi_output_sub-element_delimiter) ◆ EDI Output Release Character (edi_output_release_character) ◆ EDI Output Repeating Element Separator (edi_output_repeating_element_delimiter) ◆ EDI Output Decimal Character (edi_output_decimal_separator) 	<p>Values to change delimiters specified in the map, if the output side of the map is in EDI format. Optional.</p>
Map Name (map_name)	Map name used by the translator.
Output Report To Process Data (output_report_to_process_data)	<p>Whether to output the report to process data. Valid values are:</p> <ul style="list-style-type: none"> ◆ Yes: Output the report to process data. ◆ No: Do no output the report to process data.
Sender Identity ID (SenderIdID)	Used by map to access trading partner codelists.
Receiver Identity ID (ReceiverIdentityID)	Used by map to access trading partner codelists.

Field	Description
Run as User	Enter (or select from the list) the user ID to be associated with business process instances of this service.
Use 24 Hour Clock Display	Select to specify times for this schedule using the 24 hour clock. Leave blank to use 12 hour clock and AM and PM.
Schedule	<p>Information about scheduling the JDBC adapter configuration to run and to start the specified business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, this service does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the month Valid values are the day of the month (including the last day of the month (LDOM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. <p>Note: The Schedule field only displays as an option if you set the Start a new business process parameter to "This JDBC adapter will start a new business process".</p>

Parameters That Must Be Added in BPML

The following parameters can be used with the JDBC adapter by editing the BPML.

Parameter	Description
FromSchema	<p>Used to enable manipulation of a database schema prefix within the SQL Table/View or SQL Statement of a map. This parameter is required when overriding schema names within one or more SQL Statement fields.</p> <p>If the FromSchema and ToSchema parameters are not supplied, then no schema name substitution is performed.</p> <p>Note: The schema search/replace is case-sensitive.</p>

Parameter	Description
ToSchema	<p>Used to enable manipulation of a database schema prefix within the SQL Table/View or SQL Statement of a map.</p> <p>Note: The schema search/replace is case-sensitive.</p> <p>If the FromSchema and ToSchema parameters are not supplied, then no schema name substitution is performed.</p> <p>If the ToSchema parameter is supplied and contains a non-empty value, then any matching schema names are changed at translation time to use the supplied ToSchema schema value as follows:</p> <ul style="list-style-type: none"> ◆ For a SQL Statement, only schema names that match the FromSchema value will be substituted. The FromSchema parameter is required—otherwise, no schema values are substituted. To match and substitute more than one value pair, the FromSchema and ToSchema parameter strings can be delimited with an @ sign. For example: <pre>FromSchema="from1@from2" ToSchema="to1@to2"</pre> <p>In this example, any schema names matching "from1" are changed to "to1," and any schema names matching "from2" are changed to "to2."</p> <p>For convenience, you can supply fewer ToSchema fragments than FromSchema fragments, and when there is no corresponding ToSchema fragment, the last fragment in the ToSchema string is used. For example: <pre>FromSchema="from1@from2@from3" ToSchema="to"</pre> <p>In this example, any schema names matching "from1," "from2," or "from3" will be changed to "to."</p> </p> ◆ For a SQL Table/View, the FromSchema parameter is optional. If it is not supplied, all schema names are changed to the supplied ToSchema value. If it is supplied, the substitution occurs in the same way as it does for a SQL Statement. If the translator property <code>sql.driver.useIdentifierQuoteString</code> is set to True within <code>customer_overrides.properties</code>, then matching and substitution occurs with quoted schema names. ◆ If the ToSchema parameter is supplied but is empty (equal to "" (two double quotation marks) or ' ' (two single quotation marks)), then any matching schema names contained in the map are removed at translation time.

Setting Up a Connection to an External Database

You must set up a connection to an external database for the JDBC adapter. You can use any of the databases supported by the Application for internal use (see the System Requirements documentation on Help on the Web), or other JDBC-compliant databases, such as Sybase.

Adding New Database Pools

To define a new database pool for use by the JDBC adapter, you must add settings for the pool to the `jdbc_customer.properties.in` file, which is located in the Application `/install_dir/properties` directory.

In `jdbc_customer.properties.in`, specify the database server name, port number, database/catalog name, user ID and password. To encrypt your database password, use the `encrypt_string.sh` or `encrypt_string.cmd`

utility in the bin directory. Then place the encrypted password, prefixed by an encryption indicator, in your properties file.

Caution: There are two `jdbc_customer.properties` files: `jdbc_customer.properties.in`, which is the “template” properties file; and `jdbc_customer.properties`, which is the “packaged” properties file.

It is extremely important to ensure that you add the records to the template file, `jdbc_customer.properties.in`, not to the packaged file.

Each time you run the `setupfiles` command in the Application, all the packaged files are updated with the information contained in their template (`.in`) files. This means that if you make changes to the packaged file, `jdbc_customer.properties`, they are lost each time `setupfiles` runs. Always make changes to the template file, `jdbc_customer.properties.in`, and your changes will be maintained.

If the database you want to connect to resides on a database server type that is not the same as the Application database server type, you must also install a JDBC driver using the `install3rdparty.sh` or `install3rdparty.cmd` utility.

Select a table and column in your database to use in the test on reserve function. This function causes the Application to test the database connection using a quickly run query before attempting to use it. This function ensures that idle connections are revived. The column referenced in the query should be of the type `varchar` and should be at least five characters in length.

Connecting to an External Database

To connect to an external database:

1. Add the necessary records to the `jdbc_customer.properties.in` file found in the `/install_dir/properties` directory.

Note: If invalid data (like `ABC` or `13 . 45`) is entered in a pool setting, the setting uses its default value.

See the examples that follow this procedure for *Oracle 8i/9i* on page 745, *DB2* on page 746, *MS SQL 2000* on page 747, and *Sybase* on page 747.

The following table contains the parameters needed to add a new database pool to the `jdbc_customer.properties.in` file:

Parameter	Description
<code>databasePool.driver</code>	JDBC driver class file for the database application.
<code>databasePool.url</code>	Database location (full URL as defined by the Java JDBC standards). Note: For Oracle systems, the last segment in the URL is the Oracle SID (not the System Reference or Tnsnames entry). Note: You can locate the Java JDBC standards on the java.sun.com Web site.
<code>databasePool.user</code>	Username for logging into the database.
<code>databasePool.password</code>	Password for logging into the database.
<code>databasePool.maxconn</code>	Maximum number of database connections for the connection pool.

Parameter	Description
<i>databasePool.storedProcClassName</i>	<p>Specifies the class that handles stored procedure calls for the JDBC adapter. The following classes are used for the database types:</p> <ul style="list-style-type: none"> ◆ MSSQL, Sybase, and DB2 – com.sterlingcommerce.woodstock.util.frame.jdbc.GenericStoredProcQuery ◆ Oracle 8i/9i – com.sterlingcommerce.woodstock.util.frame.jdbc.OracleNoAppStoredProcQuery <p>Note: The JDBC adapter does not support stored procedures for DB2/iSeries, DB2/zOS, and MySQL.</p>
<i>databasePool.varDataClassName</i>	<p>Each database that the Application supports handles binary objects differently. This parameter specifies the class used to handle binary data for the database. Enter the correct class for your database:</p> <ul style="list-style-type: none"> ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ISeriesVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2VarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ZOSVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.JConnectVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.MSSQLVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.MySQLVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.OracleBlobVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.OracleVarData
<i>databasePool.catalog</i>	Database name (usually the same as the last segment of the URL)
<i>databasePool.type</i>	<p>Valid values:</p> <ul style="list-style-type: none"> ◆ local ◆ remote (default)
<i>databasePool.testOnReserve</i>	<p>Whether to test the connection.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ true ◆ false <p>Note: This function causes the Application to test the database connection before attempting to use it and revives idle connections.</p>

Parameter	Description
<i>databasePool.testOnReserveQuery</i>	<p>SQL query to use when testing the connection. Select a table and column in your database to use in the test on reserve function. The column referenced in the query should be of the type <i>varchar</i> and should be at least five characters in length. This query needs to be executable by the <i>databasePool.username</i> account and must be a valid SQL query. For example:</p> <pre>SELECT table_name FROM user_tables WHERE table_name=?</pre> <p>where ? must accept a string value. The query does not have to return a value to operate. If the query fails, the Database Pool is not activated.</p>
<i>databasePool.testOnReserveInterval</i>	<p>The minimum number of milliseconds between running <i>testOnReserve</i> on the same connection. The default value is 60000. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 - No interval and current interval is used. ◆ <= 0 - No interval. ◆ > 0 - The minimum number of milliseconds between running <i>testOnReserve</i> on the same connection.
<i>databasePool.max8177RetryCount</i>	<p>Only used for an Oracle database, this tells the software how many times to retry if it receives an ORA-8177 error in certain situations.</p>
<i>databasePool.dbvendor</i>	<p>Enter the database name: sybase, oracle, mysql, mssql, db2, db2zos, db2iseries, or other vendor name.</p>
<i>databasePool.maxsize</i>	<p>Maximum size of the database pool. This property was previously contained in the <i>poolManager.properties</i> file. This value must not exceed the value specified for the <i>databasePool.maxconn</i> parameter in the <i>jdbc.properties</i> file.</p>
<i>databasePool.initsize</i>	<p>Initial size of the database pool. This property was previously contained in the <i>poolManager.properties</i> file.</p>
<i>databasePool.factory</i>	<p>Always enter the following: com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory</p>
<i>databasePool.behaviour</i>	<p>Behavior a connection pool exhibits when it runs out of connections. This property replaces the <i>databasePool.onEmpty</i> property in the former <i>poolManager.properties</i> file. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 – The pool simply returns indicating to the software to abort its current action and try again later. This value corresponds to the value return in the <i>databasePool.onEmpty</i> property. ◆ 1 – The pool waits the number of milliseconds specified in <i>databasePool.waittime</i> for a connection to be returned before indicating to the software to abort and try again. This value corresponds to the value wait in the <i>databasePool.onEmpty</i> property. ◆ 2 – The pool creates a buffered connection (a connection above the size specified in <i>databasePool.maxsize</i>). When using a setting of 2, the maximum number of connections for the pool is the value specified for <i>databasePool.maxsize</i> plus the value specified for <i>databasePool.buffersize</i> connections. This allows connections to be created under heavy demand. This value corresponds to the value new in the <i>databasePool.onEmpty</i> property.

Parameter	Description
<code>databasePool.lifespan</code>	The number of milliseconds a connection will live in a given pool before it needs to be removed. <ul style="list-style-type: none"> ◆ 0 - (Default) No timeout. ◆ <= 0 - No timeout. ◆ > 0 - Number of milliseconds that a connection stay in pool.
<code>databasePool.idletimeout</code>	The number of milliseconds a connection can stay idle in a given pool before it needs to be removed. The default value is 86400000. Valid values: <ul style="list-style-type: none"> ◆ 0 - No timeout. ◆ <= 0 - No timeout. ◆ > 0 - Number of milliseconds that a connection stay in pool.
<code>databasePool.housekeepinginterval</code>	The minimum number of milliseconds between running the housekeeping task to clean out idle connections. Valid values are any positive number. The default value is 3600000 milliseconds (1 hour). Any number less than 3600000 will cause the default of 3600000 milliseconds to be used.
<code>databasePool.bufferSize</code>	Number of extra connections that the connection pool can create above the value specified for <code>databasePool.maxSize</code> to improve handling of unanticipated loads on the system. This property is only used if <code>databasePool.behavior</code> is set to 2.
<code>databasePool.waitTime</code>	Amount of time (in milliseconds) to wait for a connection to become available before indicating to the software to abort the current action and try again later. This property is only used if <code>databasePool.behavior</code> is set to 1.

2. Run the `setupfiles.sh` (UNIX) or `setupfiles.cmd` (Windows) utility located in the `/install_dir/bin` directory of the Application installation directory. This updates the “packaged” properties file, `jdbc_customer.properties`, with the changes from the “template” properties file, `jdbc_customer.properties.in`.
3. If the vendor for the connection database is not the same vendor as the database vendor used for the Application database, install the appropriate JDBC driver to access the database server. Use the `install3rdParty.sh` (UNIX) or `install3rdparty.cmd` (Windows) utility located in the `/install_dir/bin` directory of the Application installation directory to add the JDBC driver jar file(s). Type **install3rdParty** on the command line to get a description of the parameters you can specify.

The following examples are for a UNIX environment. The vendor name and version are the first two parameters, along with the location of the zip file containing the JDBC driver files.

- ◆ For Oracle 9i, install the driver using the following command:

```
./install3rdParty.sh Oracle 9_2_0_5 -d
/usr/local/directory
```

- ◆ For DB2, install the driver using the following command:

```
./install3rdParty.sh db2java 7_2 -d /usr/local/directory/db2java.zip
```

- ◆ To install the jConnect driver for Sybase, refer to *Installing a Sybase Driver* on page 743.

4. Stop and restart the Application to use the changed files.

Installing a Sybase Driver

Install the jConnect driver for Sybase using the following procedure:

1. Download jConnect-5_5.zip from the Sybase web site.

2. Run the following command:

```
./install3rdParty.sh jconnect 5_5 -d  
/usr/local/directory/jconnect/5_5/jConnect-5_5.zip
```

- ◆ If this command succeeds, you are finished with this procedure.
- ◆ If Application reports in the system log that the driver could not be registered because the driver class cannot be found, continue with the procedure. Use the following steps to remove existing references to jConnect.

3. Stop Application.

4. Change your directory to *install_dir*/jar.

5. Delete any existing folders referencing jConnect.

6. Change your directory to *install_dir*/properties.

7. Open the files *dynamicclasspath.cfg* and *dynamicclasspath.cfg.in*. Delete any lines referencing jConnect, and save the files.

8. Create the following temporary directory:

```
install_dir/bin/jconnect
```

9. Extract only the jar files from jConnect-5_5.zip to this directory.

10. Run the following command:

```
install3rdparty jconnect 5_5 -d install_dir
```

11. Check *install_dir*/jar/jconnect/5_5/*your_platform* to make sure that six jar files have been copied successfully.

12. Open the *dynamicclasspath.cfg* file in *install_dir*/properties and check that the following entries are there:

```
VENDOR_JAR=/                /jar/jconnect/5_5/your_platform  
                install_dir                your_platform  
                install_dir                your_platform jisql.jar  
VENDOR_JAR=/                /jar/jconnect/5_5/                /jTDS2.jar  
VENDOR_JAR=/                /jar/jconnect/5_5/                /jTDS2d.jar  
VENDOR_JAR=/                /jar/jconnect/5_5/                /ribo.jar
```

13. Open the `customer.jdbc.properties.in` file in `install_dir/properties` and check that the following entries are there:

```
jconnectPool.driver=com.sybase.jdbc2.jdbc.SybDriver
jconnectPool.url=jdbc:sybase:Tds:your Hostname:4100/your DB
jconnectPool.user=
jconnectPool.password=your password
jconnectPool.catalog=your database
```

```
jconnectPool.testOnReserveQuery=your Test On Reserve Query
```

```
jconnectPool.maxsize=28
jconnectPool.initsize=1
jconnectPool.behaviour=2
jconnectPool.waittime=1000
jconnectPool.storedProcClassName=
com.sterlingcommerce.woodstock.util.frame.jdbc.SybaseStoredProcQuery
jconnectPool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc
c.JConnectVarData
jconnectPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory
```

14. Save the `customer.jdbc.properties.in` file and run the following command:

```
/bin/setupfiles.sh
```

This procedure should result in a successful connection to your Sybase database. However, if the database has been configured as character set ROMAN8, it is likely that you will see the following message in the Application system log, because of a limitation in the Sybase driver:

```
java.sql.SQLException: JZ0IB: The server's default charset of roman8 does not map
to an encoding that is available in the client Java environment. Because jConnect
will not be able to do client-side conversion, the connection is unusable and
is being closed. Try using a later Java version or try including your Java
installation's i18n.jar or charsets.jar file in the classpath
```

One resolution of this issue is to configure the primary Adaptive Server with a default character set that maps to one of the character sets supported by jConnect for JDBC (for example, UTF-8). For more details, refer to the Sybase documentation.

Another resolution of this issue is to use the open source jTDS driver from Sourceforge (sourceforge.net). To install this driver, follow these instructions:

1. Stop Application.
2. Remove references to jConnect as described previously.
3. Copy the `jtids-1.2.jar` file to an accessible directory on the Application machine.

4. Run the following command:

```
/bin/Install3rdparty.sh jTDS 1_2 - jar absolutePath
```

5. Check that the dynamicclasspath.cfg file has picked up this change. For example, */install_dir/jar/jTDS/1_2/your_platform/jtds-1.2.jar*.
6. Edit the jdbc_customer.properties.in file. The definition of the pool should be similar to the following example:

```
your user name  
your password  
your database
```

```
#jTDSPool.testOnReserveQuery=  
jTDSPool.dbvendor=jtds  
jTDSPool.bufferSize=50  
jTDSPool.maxSize=20  
jTDSPool.initSize=5  
jTDSPool.behaviour=2  
jTDSPool.storedProcClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.  
.SybaseStoredProcQuery  
jTDSPool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.Ge  
nericVarData  
jTDSPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionF  
actory
```

7. Restart Application.

Properties File Examples for Specific Databases

Oracle 8i/9i

For Oracle 8i/9i, enter the following parameters in the jdbc_customer.properties.in file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=oracle.jdbc.OracleDriver  
databasePool.url=jdbc:oracle:thin:@servername:0000:servername  
databasePool.user=username  
databasePool.password=password  
databasePool.catalog=catalogname  
databasePool.type=local
```

databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.max8177RetryCount=n
databasePool.dbvendor=oracle
databasePool.bufferSize=n
databasePool.maxSize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idleTimeout=86400000
databasePool.housekeepingInterval=3600000
databasePool.waitTime=n
databasePool.storedProcClassName=
com.sterlingcommerce.woodstock.util.frame.jdbc.OracleNoAppStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.OracleVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory

DB2

For DB2, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

Note: The JDBC adapter does not support stored procedures for DB2/iSeries and DB2/zOS.

databasePool.driver=com.ibm.as400.access.AS400JDBCdriver
databasePool.url=jdbc:as400://serverName/DB2Database;translate binary=true;transaction isolation=none;
databasePool.catalog=DB2Database
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ISeriesVarData
databasePool.user=username
databasePool.password=password
databasePool.maxConn=20
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT SI_VERSION from SI_VERSION where SI_VERSION = ?
databasePool.testOnReserveInterval=60000
databasePool.blobPageSize=1024000
databasePool.compressBlob=true
databasePool.cachePgs=true
databasePool.dbVendor=db2iSeries
databasePool.bufferSize=500
databasePool.maxSize=20
databasePool.initsize=0
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory

databasePool.behaviour=2
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=1000

MS SQL 2000

For MS SQL 2000, enter the following parameters in the *jdbc_customer.properties.in* file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

databasePool.driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
databasePool.url=*jdbc:microsoft:sqlserver://servername:0000;DatabaseName=SQLdatabase;SelectMethod=cursor*
databasePool.user=*username*
databasePool.password=*password*
databasePool.catalog=*catalogname*
databasePool.type=local
databasePool.testOnReserve=*true*
databasePool.testOnReserveQuery=*SELECT TestConnection from Connection_tb WHERE TestConnection = ?*
databasePool.testOnReserveInterval=60000
databasePool.dbvendor=mssql
databasePool.bufferize=*n*
databasePool.maxsize=*n*
databasePool.initsize=*n*
databasePool.behaviour=*n*
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=*n*
databasePool.storedProcClassName= com.sterlingcommerce.woodstock.util.frame.jdbc.GenericStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.MSSQLVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory

Sybase

For Sybase, enter the following parameters in the *jdbc_customer.properties.in* file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

databasePool.driver=com.sybase.jdbc2.jdbc.SybDriver
databasePool.url=*jdbc:sybase:Tds:servername:0000/SybaseDB*
databasePool.user=*username*
databasePool.password=*password*

```
databasePool.catalog=catalogname
databasePool.type=local
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.dbvendor=Sybase
databasePool.bufferSize=n
databasePool.maxSize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idleTimeout=86400000
databasePool.housekeepingInterval=3600000
databasePool.waitTime=n
databasePool.storedProcClassName= com.sterlingcommerce.woodstock.util.frame.jdbc.SybaseStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.JConnectVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory
```

Encrypting Your Database Password

To encrypt your database password:

1. Use `encrypt_string.sh` (in Windows, `encrypt_string.cmd`).
2. When prompted, enter your external database password.

The script returns the encrypted value for your password.

3. Place the encrypted password in your `jdbc.properties.in` file entry (see step 2 in the previous procedure), prefixing the encrypted password with `ENCRYPTED`.

For example, `myDSN.password=ENCRYPTED:r00ABXQABkRFU2VkZXVy`

JCA Listener Service and GIS Resource Adapter

The JCA Listener service and GIS Resource adapter work together to enable you to integrate Application with your legacy systems. The following table provides an overview of the JCA Listener service:

System name	SI_JCA_ADAPTER
Graphical Process Modeler (GPM) category	None
Description	Listens for incoming requests from Java Connector Architecture™ (JCA) requests. Runs business processes and returns resultant documents. A JCA Resource adapter (for Application, called the GIS Resource adapter) is a standard mechanism in J2EE™ for supplying connectivity to remote enterprise integration systems from an application server. The JCA Listener service receives information from the GIS Resource adapter and starts a business process.
Business usage	The GIS Resource adapter is deployed in two parts: one in a remote application server, the other in Application. This enables you to use an Enterprise Java Bean (EJB) that you write to start Application business processes directly from your application server. You can also pass documents to Application for processing and retrieve documents from Application. This enables you to integrate Application into your existing systems.
Usage example	A legacy system running on an application server (for example, JBoss™ or WebLogic®) contains data such as a purchase order or bill of materials. The legacy system can run a business process in Application and pass documents into Application for processing. The legacy system can also retrieve the results of the processing from Application.
Preconfigured?	No
Requires third party files?	Application server must have j2ee.jar (v1.3.1) and xerces.jar in its classpath.
Platform availability	All supported Application platforms
Related services	The JCA Listener service is designed to work with the GIS Resource adapter for Application, which is deployed in an application server.
Application requirements	The GIS Resource adapter (v 1.0) for Application must be deployed in an application server before using the JCA Listener service.
Initiates business processes?	This adapter starts business processes requested through the JCA Listener service on the remote application server. You integrate the remote application with an EJB written to access Application through the JCA Listener service.
Invocation	Does not run by a business process.
Business process context considerations	Any text nodes placed under the message_to_child node at the top of ProcessData will be returned to the calling EJB.
Returned status values	If this adapter starts from a business process, it will return an error. The completed business process returns the process ID to the Listener, which then allows for using the PID to query for the status, and to retrieve process data or documents.

Restrictions	There may be multiple configurations of this adapter, but each must listen on a unique port.
--------------	--

Requirements

To use the GIS Resource adapter, you should have knowledge of:

- JCA specification
- Enterprise Java Beans
- XML concepts
- How process data and documents are handled in Application

For the GIS Resource adapter to work correctly, verify that:

- You have deployed the GIS Resource adapter in an external application server.
- You have installed and configured the JCA Listener service in Application.
- The files `j2ee.jar` (version 1.3.1 or higher) and `xerces.jar` are in the application server classpath.

How the GIS Resource Adapter Works

The GIS Resource adapter is deployed in a remote application server. This enables you to use a custom-written Enterprise Java Bean (EJB) to run a business process inside Application. You can also pass documents to Application for processing and retrieve documents from Application. This enables you to integrate Application into your existing systems.

The GIS Resource adapter and the JCA Listener service can be configured to run in synchronous or asynchronous mode.

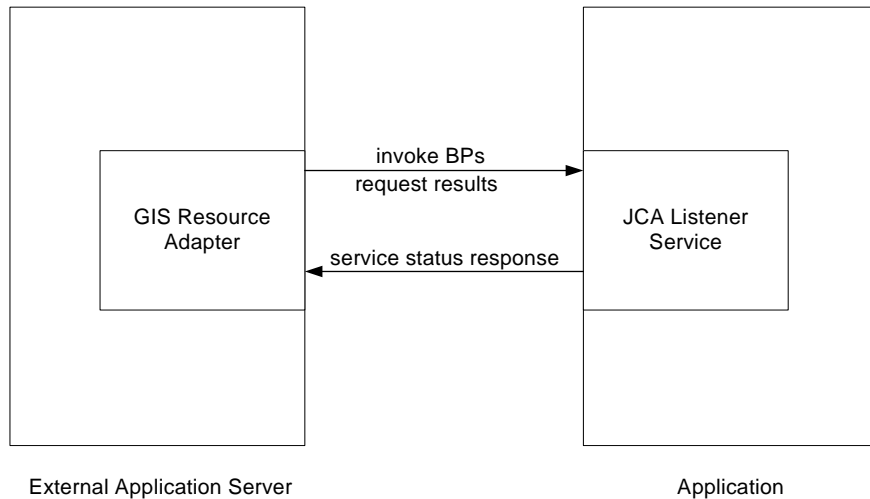
The following steps summarize how the GIS Resource adapter and the JCA Listener service work in synchronous mode. In this mode, you can run a predefined Application business process and receive the resulting document when the business process completes.

1. A legacy system running on an external application server has a document that requires processing by Application.
2. The legacy system uses an EJB you write to specify the Application business process to run and to pass the document to the GIS Resource adapter.
3. The GIS Resource adapter transfers the document and the name of the business process to run to the JCA Listener service running in Application.
4. The JCA Listener service starts the specified business process and passes it the input document.
5. When the business process is completed, the resultant document is returned to the GIS Resource adapter and back to the EJB.

Using the asynchronous mode of communication, you can run a predefined Application business process without waiting for the business process to complete. In this case, you can go back at some later time to retrieve resulting documents and status.

Note: The JCA Listener service cannot run as part of a business process.

The following figure shows how the GIS Resource adapter, running on an external application server, interacts with the JCA Listener service running on Application.



Limitations to Full JCA Specification Support

The GIS Resource adapter does not support the following portions of the JCA specification:

- Non-managed two-tier application scenarios, that is, applications that are not managed by an application server

- Transactions

- Connection pooling, in the traditional sense, as the adapter does not persist any live connections

Sample Business Scenario

This section describes a sample business scenario involving the JCA adapter.

You have a legacy billing system, and need to pass invoices to Application for processing, then receive back the results of that process.

1. Write the necessary business process in Application.
2. Configure the GIS Resource adapter to work with your legacy billing system server.
3. Write an EJB that passes the file from the legacy billing system to the GIS Resource adapter, and specifies the business process to run in Application.
4. Identify a free port on the server for the JCA Listener service to listen on.
5. Create a JCA Listener service configuration. For information, see *Managing Services and Adapters*.
6. Configure the JCA Listener service. For information, see *Configuring the JCA Listener Service* on page 753.
7. Enable the JCA Listener service and GIS Resource adapter.

8. When your billing system produces a file to be sent to Application, the EJB picks up the file from the legacy system and passes it and the business process name to the GIS Resource adapter.
9. The GIS Resource adapter passes the file and business process name to the JCA Listener service.
10. The business process starts, runs successfully, and puts the results in the business process context.
11. The response is passed back from the JCA Listener service to the GIS Resource adapter, then to the legacy system.

Implementing the GIS Resource Adapter

To implement the GIS Resource adapter, complete the following tasks:

1. Deploy the GIS Resource adapter on your application server.
2. Configure the GIS Resource adapter using your application server administrator console or by editing the application server deployment descriptors.
3. Make classpath changes to support compiling external J2EE components.
4. Create a JCA Listener service configuration. For information, see *Managing Services and Adapters*.
5. Configure the JCA Listener service. For information, see *Configuring the JCA Listener Service* on page 753.

You can configure multiple instances of this adapter, each one on a unique port.

6. Use the JCA Listener service in a business process.

Installed Location

The resource adapter (RAR) file containing the resource adapter implementation and the deployment descriptors is available under the client/jca subdirectory. The RAR files for deploying in JBoss and in WebLogic application servers are under the JBoss and WebLogic subdirectories, respectively.

JBoss Deployment

To deploy the GIS Resource adapter on a JBoss application server, copy the following files to the application server deployment directory:

sijca-service.xml – JBoss deployment descriptor for the resource adapter.

sijca_<version>_jboss.rar – Contains the resource adapter implementation classes and the standard deployment descriptor, ra.xml.

The Resource adapter can then be accessed through the JBoss JMX console.

WebLogic Deployment

To deploy the GIS Resource adapter in a WebLogic application server, the resource adapter (RAR file) must be packaged in an Enterprise Archive (EAR) file along with the J2EE component that will access it. It must be deployed as an EAR application to establish that the CCI classes are loaded and accessible to the J2EE component.

The sijca_<version>_wl.rar contains:

- ra.xml – The standard deployment descriptor for the Resource adapter
- weblogic-ra.xml – The WebLogic deployment descriptor for the Resource adapter
- Resource adapter implementation classes

Alternatively, `si_jca_<version>_client.jar` can be added to the WebLogic system classpath. In that case, deployment of new versions of the resource adapter will require an application server restart. To leverage the hot-deploy capability of the WebLogic server, it is recommended that the Resource adapter be deployed as part of an EAR application.

The Resource adapter can be accessed and configured through the WebLogic server administrator console.

Configuration

Before you can use the GIS Resource adapter, you must configure the following parameters:

- HostName – The host machine where the targeted Application instance is running
- PortNumber – The port number where the JCA Listener service configuration is running

These parameters can be configured through the application server administrator consoles, or by editing the Resource adapter deployment descriptors (`sijca-service.xml` for JBoss, and `weblogic-ra.xml` for WebLogic).

The user name and password for authenticating the user must be provided through the `ConnectionSpec` when starting `getConnection()` on the `ConnectionFactory`.

Classpath Changes

To compile the external J2EE components that will access the GIS Resource adapter, `si_jca_<version>_client.jar` (under `client/jca`) has to be added to the classpath. The javadoc for these CCI (Common Client Interface) classes is available under the `client/jca/javadoc` subdirectory.

Configuring the JCA Listener Service

To configure the JCA Listener service, you must specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>

Field	Description
Listen Port (listenPort)	Port number that this instance of the adapter will listen on. This port must be a free port. Numeric. Valid values are 1025 - 65535. Required.

Functions Supported for Application

The GIS Resource adapter enables you to use Application functionality, such as:

Running a business process with or without an input document, and with or without input process data

To run a business process in Application:

- a. Set the operationName in the SIInteractionSpec to InvokeBusinessProcess.
- b. To specify synchronous or asynchronous invocation, set the operationMode in the InteractionSpec as follows:

To run the business process synchronously, set the operationMode to InteractionSpec.SYNC_SEND_RECEIVE

To run the business process asynchronously, set the operationMode to InteractionSpec.SYNC_SEND

Retrieving documents and output process data from completed business processes

Any process data parameters that are included inside a message_to_child xml tag in process data of the business process that is running are returned as output process data. In the following example, param1 and param2 are returned as output process data, but not param3.

```
<process_data>
  <message_to_child>
    <param1>ValueOfParam1</param1>
    <param2>ValueOfParam2</param2>
  </message_to_child>
  ...
  <param3>ValueOfParam3</param3>
  ...
</process_data>
```

To retrieve a document and/or output process data from a completed business process, set the operationName to GetBusinessProcessContext; operationMode is disregarded for this operation.

Retrieving the state of a business process.

To retrieve the state of a business process, set the operationName to GetBusinessProcessState; operationMode is disregarded for this operation.

Code Example

A sample EJB illustrating the use of the GIS Resource adapter is available under the samples/jca directory.

1. Look up the Resource Adapter Connection Factory:

```
InitialContext iniCtx = new InitialContext();
    /*
    // For JBoss Object ref = iniCtx.lookup("java:/SI");
    */
    // For WebLogic
    Object ref = iniCtx.lookup("eis/SI");
    System.out.println("Found ConnectionFactory - " +
ref.getClass().getName());
    cf = (SIConnectionFactory) ref;
    System.out.println("Found ConnectionFactory");
```

2. Make a connection. At this point, the GIS Resource adapter connects to the JCA Listener service on Application, authenticates the user, and disconnects. The user name and password come from the Application Accounts area:

```
SIConnectionSpec info = new SIConnectionSpec("admin", "password");
    Connection conn = cf.getConnection(info);
    System.out.println("Created Connection #1");
    Interaction interaction = conn.createInteraction();
```

3. Create an InteractionSpec defining the operation to be run and the mode of operation.

```
InteractionSpec interSpec = new SIInteractionSpec(
    SIInteractionSpec.OP_INVOKE_BP,
    InteractionSpec.SYNC_SEND_RECEIVE);
```

4. Create a MappedRecord and set the required input data.

```
MappedRecord inRec = new SIMappedRecord();
    inRec.setRecordName("InputRecord");
    inRec.setRecordShortDescription("Input record");
```

5. Set the following for passing an input document:

```
inRec.put(SIMappedRecordKeys.DOC_NAME, "DocName");
    inRec.put(SIMappedRecordKeys.DOC_SUBJECT, "DocSubject");
    inRec.put(SIMappedRecordKeys.DOC_TYPE, "text");
    inRec.put(SIMappedRecordKeys.DOC_SUBTYPE, "plain");
    inRec.put(SIMappedRecordKeys.DOC_ENCODING, "UTF-8");
    String docBody = new String("This is the body of the test
document");
    inRec.put(SIMappedRecordKeys.DOC_BODY, docBody.getBytes());
```

6. Pass any input process data as follows:

```
Hashtable processData = new Hashtable();
    processData.put("processData1", "process data value 1");
    processData.put("processData2", "process data value 2");
    inRec.put(SIMappedRecordKeys.PROCESS_DATA, processData);
```

7. Set the name of the business process to run.

```
inRec.put(SIMappedRecordKeys.INVOKE_BP_NAME, "BoomerangJCA");
```

8. Execute the operation.

```
MappedRecord outRec = (MappedRecord)interaction.execute(interSpec, inRec);
```

9. Retrieve the output process data. Only parameters that are inside a message_to_child node in process data will be returned.

```
Hashtable pd = (Hashtable) outRec.get(SIMappedRecordKeys.PROCESS_DATA);
    Enumeration pdKeys = pd.keys();
    while (pdKeys.hasMoreElements()) {
        System.out.println("Process Data key = " +
pdKeys.nextElement());
    };
```

10. Release the connection.

```
interaction.close();
    conn.close();
```

JMS Queue Adapter (Build 4300 - Build 4317)

The following table provides an overview of the JMS Queue adapter:

System name	JMS Queue Adapter
Graphical Process Modeler (GPM) category	All Services and Messaging > Queuing
Description	Exchanges messages with remote JMS Queues. Use this adapter when you want to send messages to or receive messages from a remote JMS Queue server as part of a business process within Application.
Preconfigured?	No
Requires third party files?	A 3rd party jar file may be necessary if the value specified for either the <i>InitJndiFactory</i> parameter or the <i>Factory</i> parameter refers to a class that is not already included in the Application installation. For example, if your application server is JBoss but you need to communicate with an external Weblogic JMS server, you need to install the jar file that includes the <i>weblogic.jndi.WLInitialContextFactory</i> class. You can obtain the necessary jar file from the corresponding vendor or your trading partner.
Platform availability	All supported Application platforms
Related services	JMS Topic adapter
Application requirements	No
Initiates business processes?	Initiates a business process when configured for async receive.
Invocation	This adapter can only be used in a business process when configured for sending or sync receive.

How the JMS Queue Adapter Works

The JMS Queue adapter is a *stateful* adapter; therefore, once the adapter is started, it establishes and maintains the connection to the configured Queue. The adapter can be configured to work in one of three modes: send, sync receive, or async receive.

Send Mode

When configured for Send mode, the adapter waits to be invoked by a business process. The adapter can either send a single workflow document in one invocation or it can send multiple workflow documents in one invocation (batch mode). Each workflow document is sent as a separate message. See *Invoking Batch Sending* on page 763.

If connection to the JMS Server is lost, JMS Queue adapter attempts to reestablish connection with the JMS Server with a retry delay of 60 seconds (60000 milliseconds) between two attempts. JMS Queue adapter attempts a maximum of twenty times to reestablish connection with the JMS Server.

Sync Receive

When configured for Sync Receive mode, the adapter waits to be invoked by a business process. Unlike during Async Receive mode, messages remain on the server until this adapter is invoked to receive the data. One advantage of using Sync Receive mode is that multiple messages can be received in one invocation of the adapter (batch mode). The number of messages received in one invocation can be limited, if necessary. Each message received is placed into the current workflow as a separate document. See *Invoking Batch Receiving* on page 764.

Async Receive

When configured for Async Receive mode, the adapter cannot be invoked by a business process. When the adapter starts and the session is established it registers an asynchronous callback listener so that messages are received when they become available. Additionally, a new workflow is started (bootstrapped) to process each message. See *Invoking Batch Receiving* on page 764.

Implementing the JMS Queue Adapter

To implement the JMS Queue adapter, complete the following tasks:

1. Activate your license for the JMS Queue adapter.
2. Set up a queue in your JMS server.
3. Create a JMS Queue adapter configuration. See *Creating a Service Configuration*.
4. Configure the JMS Queue adapter. See *Configuring the JMS Queue Adapter* on page 758.
5. Create a business process that includes the JMS Queue adapter and enable it.
6. Test the business process and the adapter.
7. Run the business process.

Configuring the JMS Queue Adapter

To configure the JMS Queue adapter, you must specify field settings in Application:

Application Configuration

The following table describes the fields used to configure the JMS Queue adapter in Application:

Note: The field names in parentheses represent the corresponding field names in the Graphical Process Modeler. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
Initial Context Factory (InitJndiFactory)	Initial context factory for connecting to the remote JMS Queue server. Used for JNDI lookup. For example, weblogic.jndi.WLInitialContextFactory. Required.
URL (JndiUrl)	Uniform Resource Locator of the application server that listens for connection requests. Required.
Remote Queue name (RemoteQueueTopicName)	Name of the remote JMS Queue that you want to exchange messages with. Required.
Remote Queue Connection Factory (Factory)	Encapsulates connection configuration information and enables JMS applications to create a connection with predefined attributes. Defines and configures one or more connection factories, and the JMS server adds them to the JNDI space during startup. The default is <code>javax.jms.QueueConnectionFactory</code> . Required.
Remote User Name (Username)	User name for accessing the JMS Server. Required if the JMS Server requires security credentials.
Remote Password (Password)	Password for accessing the JMS Server. Required if the JMS Server requires security credentials.
Turn on debug messages (Debug)	<p>Whether to log debug messages for this adapter instance. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Debug messages will be logged. ◆ No – Debug messages will not be logged.
Queue Type (Action)	<p>Type of queue to access. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ Queue Send – Send messages. ◆ Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message. ◆ Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Field	Description
Message Type (Payload)	Type of message to send. Used only if queue type is Queue Send. Valid values are: <ul style="list-style-type: none"> ◆ BytesMessage ◆ ObjectMessage ◆ StreamMessage ◆ TextMessage
Bootstrap Workflow (InitialWorkFlowId)	Business process to initiate when data is received. Used only if queue type is Queue Receive Async. Required.
Document Storage Type (docStorageType)	Defines how the document will be stored in the system. Used only if queue type is Queue Receive Async. Required. Valid values: <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Note: See <i>Selecting a Document Storage Method for Bootstrap Adapters</i>.</p>
Maximum Bootstrap Threads (MaxThreads)	Maximum number of threads used when receiving files and starting business processes. Used only if queue type is Queue Receive Async. Each message received uses one thread. Default is 10. Optional.
Buffer Size (BufferSize)	Size of the buffer when receiving data. Used only if queue type is Queue Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional.
Document Filename (OutputFileName)	If you choose Queue Receive Async as the queue type for the JMS Queue adapter, then you can specify a file name for the data that the JMS Queue receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form <nodename>_yyyymmddhhmmsslll.
Connection retry attempts (RetryCount)	Maximum number of connection retry attempts. Used only if queue type is Queue Receive Async. Specify -1 for an infinite number of retry attempts. Default is 20. Optional.
Delay between retries (RetrySleep)	Number of milliseconds to wait between retry attempts. Default is 300000 ms (5 minutes). Used only if queue type is Queue Receive Async. Optional.
Notification Workflow (NotifyWorkFlow)	Business process initiated by the JMS Queue adapter if the maximum number of connection retries specified in <i>Connection retry attempts</i> is exceeded. Used only if queue type is Queue Receive Async. Required. If the adapter does not initiate a business process, select Not Applicable.

Field	Description
User	User ID to use for running the adapter. Select a user ID from the list. Valid values: Any valid Application user ID Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.
Jar Locations	Optional. Specify the preferred libraries of the jar files to be loaded with the JMS Queue adapter. You must specify the full path of the location of the jar files. Use semicolon (;) to separate multiple paths.

Graphical Process Modeler Configuration

For the JMS Queue adapter, there are no fields required to be configured in the GPM.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the JMS Queue service:

Parameter	Description
batchSndFilter	Optional. Only used when sending. If specified in the business process, triggers batch mode sending based on the documents that match the filter. You can use an asterisk '*' in the filter as a wildcard.
batchRcvLimit	Optional. Only used when receiving synchronously. If specified in the business process, the number of messages received is limited to the number specified. If not specified, all messages available are received.
batchRcvTimeout	Optional. Only used when receiving synchronously. If specified in the business process, it overrides the default receive timeout. If not specified, the default timeout is 2000 milliseconds (2 seconds).

Setting JMS Header Object Properties

When sending, you can set JMS object properties within the JMS header that are not part of the payload data. You can specify name/value pairs during runtime within the BPML. Because the user defined name/value pairs are unknown ahead of time, they cannot be set in the Application or GPM configuration so they must be manually added directly in the BPML. The JMS Queue adapter will look in ProcessData for the XML node name JMSetProperty and use any child nodes it finds to set the name/value pairs. There is a list of reserved property names that will set specific JMS message properties. An example of the ProcessData XML tree would look like this:

```
<ProcessData>
  <JMSetProperty>
    <somename1>somevalue1</somename1>
```

```
<somename2>somevalue2</somename2>
```

Reserved names that set specific JMS message properties

```
<correlationID>someStringValue</correlationID >
<deliveryMode>someIntegerValue</deliveryMode>
<destination>someQueueName</destination>
<expiration>someLongValue</expiration>
<messageID>someStringValue</messageID>
<priority>someIntegerValue</priority>
<redelivered>someBooleanValue(true/false)</redelivered>
<replyTo>someQueueName</replyTo>
<timestamp>someLongValue</timestamp>
<type>someStringValue</type>
</JMSetProperty>
</ProcessData>
```

An example of BPML that could be used to set these ProcessData name/value pairs follows:

```
<assign to="JMSetProperty/somename1" from=" 'somevalue1' " append="true"/>
<assign to="JMSetProperty/somename2" from=" 'somevalue2' " append="true"/>
```

When receiving, the JMS Queue adapter will set ProcessData items for all the JMS header fields and any object properties. Any object properties set in the JMS header will be put into ProcessData with the node name of JMS. For example, if there is a property called *somename* with a value of *somevalue*, ProcessData will contain JMS/somename with the corresponding value:

```
<JMS>
  <somename>somevalue</somename>
</JMS>
```

In addition to the user defined properties, the JMS Queue adapter will also set the following JMS header fields in ProcessData (if they are not null):

- JMS/correlationID
- JMS/deliveryMode
- JMS/destination
- JMS/expiration
- JMS/messageID
- JMS/priority
- JMS/redelivered
- JMS/replyTo
- JMS/timestamp
- JMS/type

The JMSetProperty can be used as a global property (under the ProcessData node) or a local property (under individual documents). Local JMSetProperty parameters override any global parameters and are useful when sending in batch mode. In the the below example, the global JMSetProperty has a parameter called “test” with a value of zero. Since the PrimaryDocument does not have a local JMSetProperty, it uses the

global one. However, since doc1, doc2, and doc3 have local JMSetProperty parameters, they use the local parameters.

```
<ProcessData>
  <JMSetProperty>
    <test>0</test>
  </JMSetProperty>
  <PrimaryDocument SCIObjectID="1833955:1063b363ed5:-774a" />
  <doc1 SCIObjectID="1833955:1063b363ed5:-774b">
    <JMSetProperty>
      <test>1</test>
    </JMSetProperty>
  </doc1>
  <doc2 SCIObjectID="1833955:1063b363ed5:-774c">
    <JMSetProperty>
      <test>2</test>
    </JMSetProperty>
  </doc2>
  <doc3 SCIObjectID="1833955:1063b363ed5:-774d">
    <JMSetProperty>
      <test>3</test>
    </JMSetProperty>
  </doc3>
</ProcessData>
```

Invoking Batch Sending

If a business process contains multiple documents in ProcessData, the JMS adapter can be invoked once with the workflow parameter **batchSndFilter**, which enables the adapter to send multiple messages for each of the documents that match the batchSndFilter criteria.

To invoke batch sending:

1. You do not need to make changes to the main adapter configuration; just add the appropriate assignment to the business process in the JMS adapter invocation step.

An example ProcessData for the example BPMLs below would look like this:

```
<ProcessData>
  <PrimaryDocument SCIObjectID="fe64b9:1060cac437b:-6a2a" />
  <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2b" />
  <XYZ>
    <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2c" />
    <doc2 SCIObjectID="fe64b9:1060cac437b:-6a2d" />
    <doc3 SCIObjectID="fe64b9:1060cac437b:-6a2e" />
  </XYZ>
</ProcessData>
```

Example 1

Sends all documents in ProcessData (including the PrimaryDocument). In this example, all five documents in ProcessData above are sent.

```
<operation name="JMS batch send">
  <participant name="JMSadapter" />
  <output message="toService">
```

```

    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'*'" />
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

```

Example 2

Sends all documents that begin with “doc” under the XYZ node. In this example, only three documents in the ProcessData above are sent.

```

<operation name="JMS batch send">
  <participant name="JMSadaptor" />
  <output message="toService">
    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'XYZ/doc*'" />
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

```

Invoking Batch Receiving

The type of receive adapter you choose to use is based on your business needs. If you are processing a large volume of messages, you may find that batching them is more efficient than bootstrapping one workflow for every message.

For the JMS Queue adapter, there are two types of receive queues:

Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message.

Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Additionally, there are two business process parameters associated with Sync Receive (batch receive):

batchRcvLimit – Optional—if used, this parameter limits the number of messages batched into the bootstrapped workflow. Default = no limit.

batchRcvTimeout – Optional—if used, this parameter specifies how long the adapter waits without receiving a message before ending. Default = 2000 (milliseconds)

Once a Sync Receive adapter completes the receive process, it creates the following information in ProcessData for the current workflow that invoked the adapter:

JMS/DocumentCount – This parameter is always created to show how many documents were created from messages received, even if zero messages were received.

JMS/Documentxxx – For every message received, a document is created under the JMS node and then sequentially numbered starting with one (that is, Document1, Document2, and so forth).

Another difference between Async Receive mode and Sync Receive mode is where the message metadata is stored in ProcessData. In Async Receive mode, it only creates one document (the PrimaryDocument), so all the metadata is stored as JMS/<metadataName>. However, in Sync Receive mode, the metadata is stored under each document as JMS/<documentName>/<metadataName> as shown in the example below.

Example of ProcessData after a batch receive was performed:

```
<ProcessData>
  <JMS>
    <DocumentCount>3</DocumentCount>
    <Document1 SCIObjectID="1833955:1060de6d03d:-697b">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156360801</messageID>
      <priority>4</priority>
      <timestamp>1125515636080</timestamp>
    </Document1>
    <Document2 SCIObjectID="1833955:1060de6d03d:-6978">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361102</messageID>
      <priority>4</priority>
      <timestamp>1125515636110</timestamp>
    </Document2>
    <Document3 SCIObjectID="1833955:1060de6d03d:-6975">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361243</messageID>
      <priority>4</priority>
      <timestamp>1125515636124</timestamp>
    </Document3>
  </JMS>
</ProcessData>
```

JMS Queue Adapter (Build 4318 or higher)

The following table provides an overview of the JMS Queue adapter:

System name	JMS Queue Adapter
Graphical Process Modeler (GPM) category	All Services and Messaging > Queuing
Description	Exchanges messages with remote JMS Queues. Use this adapter when you want to send messages to or receive messages from a remote JMS Queue server as part of a business process within Application.
Preconfigured?	No
Requires third party files?	A 3rd party jar file may be necessary if the value specified for either the <i>InitJndiFactory</i> parameter or the <i>Factory</i> parameter refers to a class that is not already included in the Application installation. For example, if your application server is JBoss but you need to communicate with an external Weblogic JMS server, you need to install the jar file that includes the <i>weblogic.jndi.WLInitialContextFactory</i> class. You can obtain the necessary jar file from the corresponding vendor or your trading partner.
Platform availability	All supported Application platforms
Related services	JMS Topic adapter
Application requirements	No
Initiates business processes?	Initiates a business process when configured for async receive.
Invocation	This adapter can only be used in a business process when configured for sending or sync receive.

How the JMS Queue Adapter Works

The JMS Queue adapter is a *stateful* adapter; therefore, once the adapter is started, it establishes and maintains the connection to the configured Queue. The adapter can be configured to work in one of three modes: send, sync receive, or async receive.

Send Mode

When configured for Send mode, the adapter waits to be invoked by a business process. The adapter can either send a single workflow document in one invocation or it can send multiple workflow documents in one invocation (batch mode). Each workflow document is sent as a separate message. See *Invoking Batch Sending* on page 772.

If connection to the JMS Server is lost, JMS Queue adapter attempts to reestablish connection with the JMS Server with a retry delay of 60 seconds (60000 milliseconds) between two attempts. JMS Queue adapter attempts a maximum of twenty times to reestablish connection with the JMS Server.

Sync Receive

When configured for Sync Receive mode, the adapter waits to be invoked by a business process. Unlike during Async Receive mode, messages remain on the server until this adapter is invoked to receive the data. One advantage of using Sync Receive mode is that multiple messages can be received in one invocation of the adapter (batch mode). The number of messages received in one invocation can be limited, if necessary. Each message received is placed into the current workflow as a separate document. See *Invoking Batch Receiving* on page 773.

Async Receive

When configured for Async Receive mode, the adapter cannot be invoked by a business process. When the adapter starts and the session is established it registers an asynchronous callback listener so that messages are received when they become available. Additionally, a new workflow is started (bootstrapped) to process each message. See *Invoking Batch Receiving* on page 773.

Implementing the JMS Queue Adapter

To implement the JMS Queue adapter, complete the following tasks:

1. Activate your license for the JMS Queue adapter.
2. Set up a queue in your JMS server.
3. Create a JMS Queue adapter configuration. See *Creating a Service Configuration*.
4. Configure the JMS Queue adapter. See *Configuring the JMS Queue Adapter* on page 767.
5. Create a business process that includes the JMS Queue adapter and enable it.
6. Test the business process and the adapter.
7. Run the business process.

Configuring the JMS Queue Adapter

To configure the JMS Queue adapter, you must specify field settings in Application:

Application Configuration

The following table describes the fields used to configure the JMS Queue adapter in Application:

Note: The field names in parentheses represent the corresponding field names in the Graphical Process Modeler. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
Initial Context Factory (InitJndiFactory)	Initial context factory for connecting to the remote JMS Queue server. Used for JNDI lookup. For example, weblogic.jndi.WLInitialContextFactory. Required.
URL (JndiUrl)	Uniform Resource Locator of the application server that listens for connection requests. Required.
Remote Queue name (RemoteQueueTopicName)	Name of the remote JMS Queue that you want to exchange messages with. Required.
Remote Queue Connection Factory (Factory)	Encapsulates connection configuration information and enables JMS applications to create a connection with predefined attributes. Defines and configures one or more connection factories, and the JMS server adds them to the JNDI space during startup. The default is javax.jms.QueueConnectionFactory. Required.
Remote User Name (Username)	User name for accessing the JMS Server. Required if the JMS Server requires security credentials.
Remote Password (Password)	Password for accessing the JMS Server. Required if the JMS Server requires security credentials.
Turn on debug messages (Debug)	<p>Whether to log debug messages for this adapter instance. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Debug messages will be logged. ◆ No – Debug messages will not be logged.
Queue Type (Action)	<p>Type of queue to access. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ Queue Send – Send messages. ◆ Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message. ◆ Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Field	Description
Message Type (Payload)	Type of message to send. Used only if queue type is Queue Send. Valid values are: <ul style="list-style-type: none"> ◆ BytesMessage ◆ ObjectMessage ◆ StreamMessage ◆ TextMessage
Bootstrap Workflow (InitialWorkFlowId)	Business process to initiate when data is received. Used only if queue type is Queue Receive Async. Required.
Document Storage Type (docStorageType)	Defines how the document will be stored in the system. Used only if queue type is Queue Receive Async. Required. Valid values: <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Note: See <i>Selecting a Document Storage Method for Bootstrap Adapters</i>.</p>
Maximum Bootstrap Threads (MaxThreads)	Maximum number of threads used when receiving files and starting business processes. Used only if queue type is Queue Receive Async. Each message received uses one thread. Default is 10. Optional.
Buffer Size (BufferSize)	Size of the buffer when receiving data. Used only if queue type is Queue Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional.
Document Filename (OutputFileName)	If you choose Queue Receive Async as the queue type for the JMS Queue adapter, then you can specify a file name for the data that the JMS Queue receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form <nodename>_yyyymmddhhmmsslll.
Connection retry attempts (RetryCount)	Maximum number of connection retry attempts. Used only if queue type is Queue Receive Async. Specify -1 for an infinite number of retry attempts. Default is 20. Optional.
Delay between retries (RetrySleep)	Number of milliseconds to wait between retry attempts. Default is 300000 ms (5 minutes). Used only if queue type is Queue Receive Async. Optional.
Notification Workflow (NotifyWorkFlow)	Business process initiated by the JMS Queue adapter if the maximum number of connection retries specified in <i>Connection retry attempts</i> is exceeded. Used only if queue type is Queue Receive Async. Required. If the adapter does not initiate a business process, select Not Applicable.

Field	Description
User	User ID to use for running the adapter. Select a user ID from the list. Valid values: Any valid Application user ID Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.
Jar Locations	Optional. Specify the preferred libraries of the jar files to be loaded with the JMS Queue adapter. You must specify the full path of the location of the jar files. Use semicolon (;) to separate multiple paths.

Graphical Process Modeler Configuration

For the JMS Queue adapter, there are no fields required to be configured in the GPM.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the JMS Queue service:

Parameter	Description
batchSndFilter	Optional. Only used when sending. If specified in the business process, triggers batch mode sending based on the documents that match the filter. You can use an asterisk '*' in the filter as a wildcard.
batchRcvLimit	Optional. Only used when receiving synchronously. If specified in the business process, the number of messages received is limited to the number specified. If not specified, all messages available are received.
batchRcvTimeout	Optional. Only used when receiving synchronously. If specified in the business process, it overrides the default receive timeout. If not specified, the default timeout is 2000 milliseconds (2 seconds).

Setting JMS Header Object Properties

When sending, you can set JMS object properties within the JMS header that are not part of the payload data. You can specify name/value pairs during runtime within the BPML. Because the user defined name/value pairs are unknown ahead of time, they cannot be set in the Application or GPM configuration so they must be manually added directly in the BPML. The JMS Queue adapter will look in ProcessData for the XML node name JMSetProperty and use any child nodes it finds to set the name/value pairs. There is a list of reserved property names that will set specific JMS message properties. An example of the ProcessData XML tree would look like this:

```
<ProcessData>
  <JMSetProperty>
    <somename1>somevalue1</somename1>
```

```
<somename2>somevalue2</somename2>
```

Reserved names that set specific JMS message properties

```
<correlationID>someStringValue</correlationID >
<deliveryMode>someIntegerValue</deliveryMode>
<destination>someQueueName</destination>
<expiration>someLongValue</expiration>
<messageID>someStringValue</messageID>
<priority>someIntegerValue</priority>
<redelivered>someBooleanValue(true/false)</redelivered>
<replyTo>someQueueName</replyTo>
<timestamp>someLongValue</timestamp>
<type>someStringValue</type>
</JMSetProperty>
</ProcessData>
```

An example of BPML that could be used to set these ProcessData name/value pairs follows:

```
<assign to="JMSetProperty/somename1" from=" 'somevalue1' " append="true"/>
<assign to="JMSetProperty/somename2" from=" 'somevalue2' " append="true"/>
```

When receiving, the JMS Queue adapter will set ProcessData items for all the JMS header fields and any object properties. Any object properties set in the JMS header will be put into ProcessData with the node name of JMS. For example, if there is a property called *somename* with a value of *somevalue*, ProcessData will contain JMS/somename with the corresponding value:

```
<JMS>
  <somename>somevalue</somename>
</JMS>
```

In addition to the user defined properties, the JMS Queue adapter will also set the following JMS header fields in ProcessData (if they are not null):

- JMS/correlationID
- JMS/deliveryMode
- JMS/destination
- JMS/expiration
- JMS/messageID
- JMS/priority
- JMS/redelivered
- JMS/replyTo
- JMS/timestamp
- JMS/type

The JMSetProperty can be used as a global property (under the ProcessData node) or a local property (under individual documents). Local JMSetProperty parameters override any global parameters and are useful when sending in batch mode. In the the below example, the global JMSetProperty has a parameter called “test” with a value of zero. Since the PrimaryDocument does not have a local JMSetProperty, it uses the

global one. However, since doc1, doc2, and doc3 have local JMSetProperty parameters, they use the local parameters.

```
<ProcessData>
  <JMSetProperty>
    <test>0</test>
  </JMSetProperty>
  <PrimaryDocument SCIObjectID="1833955:1063b363ed5:-774a" />
  <doc1 SCIObjectID="1833955:1063b363ed5:-774b">
    <JMSetProperty>
      <test>1</test>
    </JMSetProperty>
  </doc1>
  <doc2 SCIObjectID="1833955:1063b363ed5:-774c">
    <JMSetProperty>
      <test>2</test>
    </JMSetProperty>
  </doc2>
  <doc3 SCIObjectID="1833955:1063b363ed5:-774d">
    <JMSetProperty>
      <test>3</test>
    </JMSetProperty>
  </doc3>
</ProcessData>
```

Invoking Batch Sending

If a business process contains multiple documents in ProcessData, the JMS adapter can be invoked once with the workflow parameter **batchSndFilter**, which enables the adapter to send multiple messages for each of the documents that match the batchSndFilter criteria.

To invoke batch sending:

1. You do not need to make changes to the main adapter configuration; just add the appropriate assignment to the business process in the JMS adapter invocation step.

An example ProcessData for the example BPMLs below would look like this:

```
<ProcessData>
  <PrimaryDocument SCIObjectID="fe64b9:1060cac437b:-6a2a" />
  <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2b" />
  <XYZ>
    <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2c" />
    <doc2 SCIObjectID="fe64b9:1060cac437b:-6a2d" />
    <doc3 SCIObjectID="fe64b9:1060cac437b:-6a2e" />
  </XYZ>
</ProcessData>
```

Example 1

Sends all documents in ProcessData (including the PrimaryDocument). In this example, all five documents in ProcessData above are sent.

```
<operation name="JMS batch send">
  <participant name="JMSadapter" />
  <output message="toService">
```

```

    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'*'" />
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

```

Example 2

Sends all documents that begin with “doc” under the XYZ node. In this example, only three documents in the ProcessData above are sent.

```

<operation name="JMS batch send">
  <participant name="JMSadaptor" />
  <output message="toService">
    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'XYZ/doc*'" />
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

```

Invoking Batch Receiving

The type of receive adapter you choose to use is based on your business needs. If you are processing a large volume of messages, you may find that batching them is more efficient than bootstrapping one workflow for every message.

For the JMS Queue adapter, there are two types of receive queues:

Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message.

Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Additionally, there are two business process parameters associated with Sync Receive (batch receive):

batchRcvLimit – Optional—if used, this parameter limits the number of messages batched into the bootstrapped workflow. Default = no limit.

batchRcvTimeout – Optional—if used, this parameter specifies how long the adapter waits without receiving a message before ending. Default = 2000 (milliseconds)

Once a Sync Receive adapter completes the receive process, it creates the following information in ProcessData for the current workflow that invoked the adapter:

JMS/DocumentCount – This parameter is always created to show how many documents were created from messages received, even if zero messages were received.

JMS/Documentxxx – For every message received, a document is created under the JMS node and then sequentially numbered starting with one (that is, Document1, Document2, and so forth).

Another difference between Async Receive mode and Sync Receive mode is where the message metadata is stored in ProcessData. In Async Receive mode, it only creates one document (the PrimaryDocument), so all the metadata is stored as JMS/<metadataName>. However, in Sync Receive mode, the metadata is stored under each document as JMS/<documentName>/<metadataName> as shown in the example below.

Example of ProcessData after a batch receive was performed:

```
<ProcessData>
  <JMS>
    <DocumentCount>3</DocumentCount>
    <Document1 SCIObjectID="1833955:1060de6d03d:-697b">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156360801</messageID>
      <priority>4</priority>
      <timestamp>1125515636080</timestamp>
    </Document1>
    <Document2 SCIObjectID="1833955:1060de6d03d:-6978">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361102</messageID>
      <priority>4</priority>
      <timestamp>1125515636110</timestamp>
    </Document2>
    <Document3 SCIObjectID="1833955:1060de6d03d:-6975">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361243</messageID>
      <priority>4</priority>
      <timestamp>1125515636124</timestamp>
    </Document3>
  </JMS>
</ProcessData>
```

JMS Topic Adapter (Build 4300 - Build 4317)

The following table provides an overview of the JMS Topic adapter:

System name	JMS Topic Adapter
Graphical Process Modeler (GPM) category	All Services and Messaging > Queuing
Description	Exchanges messages with remote JMS Topics. Use this adapter when you want to send messages to or receive messages from a remote JMS Topic server as part of a business process within Application.
Preconfigured?	No
Requires third party files?	A 3rd party jar file may be necessary if the value specified for either the <i>InitJndiFactory</i> parameter or the <i>Factory</i> parameter refers to a class that is not already included in the Application installation. For example, if your application server is JBoss but you need to communicate with an external Weblogic JMS server, you need to install the jar file that includes the <i>weblogic.jndi.WLInitialContextFactory</i> class. You can obtain the necessary jar file from the corresponding vendor or your trading partner.
Platform availability	All supported Application platforms
Related services	JMS Queue adapter
Application requirements	No
Initiates business processes?	Initiates a business process when configured for async receive.
Invocation	This adapter can only be used in a business process when configured for sending or sync receive.

How the JMS Topic Adapter Works

The JMS Topic adapter is a *stateful* adapter; therefore, once the adapter is started, it establishes and maintains the connection to the configured Topic. The adapter can be configured to work in one of three modes: send, sync receive, or async receive.

Send Mode

When configured for Send mode, the adapter waits to be invoked by a business process. The adapter can either send a single workflow document in one invocation or it can send multiple workflow documents in one invocation (batch mode). Each workflow document is sent as a separate message. See *Invoking Batch Sending* on page 781.

If connection to the JMS Server is lost, JMS Topic adapter attempts to reestablish connection with the JMS Server with a retry delay of 60 seconds (60000 milliseconds) between two attempts. JMS Topic adapter attempts a maximum of twenty times to reestablish connection with the JMS Server.

Sync Receive

When configured for Sync Receive mode, the adapter waits to be invoked by a business process. Unlike during Async Receive mode, messages remain on the server until this adapter is invoked to receive the data. One advantage of using Sync Receive mode is that multiple messages can be received in one invocation of the adapter (batch mode). The number of messages received in one invocation can be limited, if necessary. Each message received is placed into the current workflow as a separate document. See *Invoking Batch Receiving* on page 782.

Async Receive

When configured for Async Receive mode, the adapter cannot be invoked by a business process. When the adapter starts and the session is established it registers an asynchronous callback listener so that messages are received when they become available. Additionally, a new workflow is started (bootstrapped) to process each message. See *Invoking Batch Receiving* on page 782.

Implementing the JMS Topic Adapter

To implement the JMS Topic adapter, complete the following tasks:

1. Activate your license for the JMS Topic adapter.
2. Set up a topic in your JMS server.
3. Create a JMS Topic adapter configuration. See *Creating a Service Configuration*.
4. Configure the JMS Topic adapter. See *Configuring the JMS Topic Adapter* on page 776.
5. Create a business process that includes the JMS Topic adapter and enable it.
6. Test the business process and the adapter.
7. Run the business process.

Configuring the JMS Topic Adapter

To configure the JMS Topic adapter, you must specify field settings in Application and in the Graphical Process Modeler (GPM).

Application Configuration

The following table describes the fields used to configure the JMS Topic adapter in Application:

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
Initial Context Factory (InitJndiFactory)	Initial context factory for connecting to the remote JMS Topic server. Used for JNDI lookup. For example, weblogic.jndi.WLInitialContextFactory. Required.
URL (JndiUrl)	Uniform Resource Locator of the application server that listens for connection requests. Required.
Remote Topic name (RemoteQueueTopicName)	Name of the remote JMS Topic that you want to exchange messages with. Required.
Remote Topic Connection Factory (Factory)	Encapsulates connection configuration information and enables JMS applications to create a connection with predefined attributes. Defines and configures one or more connection factories, and the JMS server adds them to the JNDI space during startup. The default is javax.jms.TopicConnectionFactory. Required.
Remote User Name (Username)	User name for accessing the JMS Server. Required if the JMS Server requires security credentials.
Remote Password (Password)	Password for accessing the JMS Server. Required if the JMS Server requires security credentials.
Turn on debug messages (Debug)	<p>Whether to log debug messages for this adapter instance. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Debug messages will be logged. ◆ No – Debug messages will not be logged.
Topic Type (Action)	<p>Type of topic to access. Valid values are:</p> <ul style="list-style-type: none"> ◆ Topic Send – Sends messages. ◆ Topic Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message. ◆ Topic Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Field	Description
Message Type (Payload)	Type of message to send. Used only if topic type is Topic Send. Valid values are: <ul style="list-style-type: none"> ◆ ByteMessage ◆ ObjectMessage ◆ StreamMessage ◆ TextMessage
Bootstrap Workflow (InitialWorkFlowId)	Business process to initiate when data is received. Used only if topic type is Topic Receive Async. Required.
Document Storage Type (docStorageType)	Defines how the document will be stored in the system. Used only if topic type is Topic Receive Async. Required. Valid values: <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Note: See <i>Selecting a Document Storage Method for Bootstrap Adapters</i>.</p>
Maximum Bootstrap Threads (MaxThreads)	Maximum number of threads used when receiving files and starting business processes. Used only if topic type is Topic Receive Async. Each message received uses one thread. Default is 10. Optional.
Buffer Size (BufferSize)	Size of the buffer when receiving data. Used only if topic type is Topic Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional.
Document Filename (OutputFileName)	If you choose Topic Receive Async as the topic type for the JMS Topic adapter, then you can specify a file name for the data that the JMS Topic receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form <nodename>_yyyymmddhhmmsslll.
Connection retry attempts (RetryCount)	Maximum number of connection retry attempts. Used only if topic type is Topic Receive Async. Specify -1 for an infinite number of retry attempts. Default is 20. Optional.
Delay between retries (RetrySleep)	Number of milliseconds to wait between retry attempts. Default is 300000 ms (5 minutes). Used only if topic type is Topic Receive Async. Optional.
Notification Workflow (NotifyWorkFlow)	Business process initiated by the JMS Topic adapter if the maximum number of connection retries specified in <i>Connection retry attempts</i> is exceeded. Used only if topic type is Topic Receive Async. Required. If the adapter does not initiate a business process, select Not Applicable.

Field	Description
User	User ID to use for running the adapter. Select a user ID from the list. Valid values: Any valid Application user ID Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.
Jar Locations	Optional. Specify the preferred libraries of the jar files to be loaded with the JMS Topic adapter. You must specify the full path of the location of the jar files. Use semicolon (;) to separate multiple paths.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the JMS Topic service:

Parameter	Description
batchSndFilter	Optional. Only used when sending. If specified in the business process, triggers batch mode sending based on the documents that match the filter. You can use an asterisk '*' in the filter as a wildcard.
batchRcvLimit	Optional. Only used when receiving synchronously. If specified in the business process, the number of messages received is limited to the number specified. If not specified, all messages available are received.
batchRcvTimeout	Optional. Only used when receiving synchronously. If specified in the business process, it overrides the default receive timeout. If not specified, the default timeout is 2000 milliseconds (2 seconds).

Setting JMS Header Object Properties

When sending, you can set JMS object properties within the JMS header that are not part of the payload data. You can specify name/value pairs during runtime within the BPML. Because the user defined name/value pairs are unknown ahead of time, they cannot be set in the Application or GPM configuration so they must be manually added directly in the BPML. The JMS Topic adapter will look in ProcessData for the XML node name JMSetProperty and use any child nodes it finds to set the name/value pairs. There is a list of reserved property names that will set specific JMS message properties. An example of the ProcessData XML tree would look like this:

```
<ProcessData>
  <JMSetProperty>
    <somename1>somevalue1</somename1>
    <somename2>somevalue2</somename2>
  </JMSetProperty>
  <Reserved names that set specific JMS message properties>
    <correlationID>someStringValue</correlationID >
    <deliveryMode>someIntegerValue</deliveryMode>
```

```

    <destination>someTopicName</destination>
    <expiration>someLongValue</expiration>
    <messageID>someStringValue</messageID>
    <priority>someIntegerValue</priority>
    <redelivered>someBooleanValue(true/false)</redelivered>
    <replyTo>someTopicName</replyTo>
    <timestamp>someLongValue</timestamp>
    <type>someStringValue</type>
  </JMSetProperty>
</ProcessData>

```

An example of BPML that could be used to set these ProcessData name/value pairs follows:

```

<assign to="JMSetProperty/somename1" from="'somevalue1'" append="true"/>
<assign to="JMSetProperty/somename2" from="'somevalue2'" append="true"/>

```

When receiving, the JMS Topic adapter will set ProcessData items for all the JMS header fields and any object properties. Any object properties set in the JMS header will be put into ProcessData with the node name of JMS. For example, if there is a property called *somename* with a value of *somevalue*, ProcessData will contain JMS/somename with the corresponding value:

```

<JMS>
  <somename>somevalue</somename>
</JMS>

```

In addition to the user defined properties, the JMS Topic adapter will also set the following JMS header fields in ProcessData (if they are not null):

- JMS/correlationID
- JMS/deliveryMode
- JMS/destination
- JMS/expiration
- JMS/messageID
- JMS/priority
- JMS/redelivered
- JMS/replyTo
- JMS/timestamp
- JMS/type

The JMSetProperty can be used as a global property (under the ProcessData node) or a local property (under individual documents). Local JMSetProperty parameters override any global parameters and are useful when sending in batch mode. In the the below example, the global JMSetProperty has a parameter called “test” with a value of zero. Since the PrimaryDocument does not have a local JMSetProperty, it uses the global one. However, since doc1, doc2, and doc3 have local JMSetProperty parameters, they use the local parameters.

```

<ProcessData>

```

```

<JMSetProperty>
  <test>0</test>
</JMSetProperty>
<PrimaryDocument SCIObjectID="1833955:1063b363ed5:-774a" />
<doc1 SCIObjectID="1833955:1063b363ed5:-774b">
  <JMSetProperty>
    <test>1</test>
  </JMSetProperty>
</doc1>
<doc2 SCIObjectID="1833955:1063b363ed5:-774c">
  <JMSetProperty>
    <test>2</test>
  </JMSetProperty>
</doc2>
<doc3 SCIObjectID="1833955:1063b363ed5:-774d">
  <JMSetProperty>
    <test>3</test>
  </JMSetProperty>
</doc3>
</ProcessData>

```

Invoking Batch Sending

If a business process contains multiple documents in ProcessData, the JMS adapter can be invoked once with the workflow parameter **batchSndFilter**, which enables the adapter to send multiple messages for each of the documents that match the batchSndFilter criteria.

To invoke batch sending:

1. You do not need to make changes to the main adapter configuration; just add the appropriate assignment to the business process in the JMS adapter invocation step.

An example ProcessData for the example BPMLs below would look like this:

```

<ProcessData>
  <PrimaryDocument SCIObjectID="fe64b9:1060cac437b:-6a2a" />
  <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2b" />
  <XYZ>
    <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2c" />
    <doc2 SCIObjectID="fe64b9:1060cac437b:-6a2d" />
    <doc3 SCIObjectID="fe64b9:1060cac437b:-6a2e" />
  </XYZ>
</ProcessData>

```

Example 1

Sends all documents in ProcessData (including the PrimaryDocument). In this example, all five documents in ProcessData above are sent.

```

<operation name="JMS batch send">
  <participant name="JMSadapter" />
  <output message="toService">
    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'*'"/>
  </output>
  <input message="fromService">

```

```

    <assign to="." from="*" />
  </input>
</operation>

```

Example 2

Sends all documents that begin with “doc” under the XYZ node. In this example, only three documents in the ProcessData above are sent.

```

<operation name="JMS batch send">
  <participant name="JMSadapter" />
  <output message="toService">
    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'XYZ/doc*'"/>
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

```

Invoking Batch Receiving

The type of receive adapter you choose to use is based on your business needs. If you are processing a large volume of messages, you may find that batching them is more efficient than bootstrapping one workflow for every message.

For the JMS Topic adapter, there are two types of receive queues:

Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message.

Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Additionally, there are two business process parameters associated with Sync Receive (batch receive):

batchRcvLimit – Optional—if used, this parameter limits the number of messages batched into the bootstrapped workflow. Default = no limit.

batchRcvTimeout – Optional—if used, this parameter specifies how long the adapter waits without receiving a message before ending. Default = 2000 (milliseconds)

Once a Sync Receive adapter completes the receive process, it creates the following information in ProcessData for the current workflow that invoked the adapter:

JMS/DocumentCount – This parameter is always created to show how many documents were created from messages received, even if zero messages were received.

JMS/Documentxxx – For every message received, a document is created under the JMS node and then sequentially numbered starting with one (that is, Document1, Document2, and so forth).

Another difference between Async Receive mode and Sync Receive mode is where the message metadata is stored in ProcessData. In Async Receive mode, it only creates one document (the PrimaryDocument), so

all the metadata is stored as JMS/<metadataName>. However, in Sync Receive mode, the metadata is stored under each document as JMS/<documentName>/<metadataName> as shown in the example below.

Example of ProcessData after a batch receive was performed:

```
<ProcessData>
  <JMS>
    <DocumentCount>3</DocumentCount>
    <Document1 SCIObjectID="1833955:1060de6d03d:-697b">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156360801</messageID>
      <priority>4</priority>
      <timestamp>1125515636080</timestamp>
    </Document1>
    <Document2 SCIObjectID="1833955:1060de6d03d:-6978">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361102</messageID>
      <priority>4</priority>
      <timestamp>1125515636110</timestamp>
    </Document2>
    <Document3 SCIObjectID="1833955:1060de6d03d:-6975">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361243</messageID>
      <priority>4</priority>
      <timestamp>1125515636124</timestamp>
    </Document3>
  </JMS>
</ProcessData>
```

JMS Topic Adapter (Build 4318 or higher)

The following table provides an overview of the JMS Topic adapter:

System name	JMS Topic Adapter
Graphical Process Modeler (GPM) category	All Services and Messaging > Queuing
Description	Exchanges messages with remote JMS Topics. Use this adapter when you want to send messages to or receive messages from a remote JMS Topic server as part of a business process within Application.
Preconfigured?	No
Requires third party files?	A 3rd party jar file may be necessary if the value specified for either the <i>InitJndiFactory</i> parameter or the <i>Factory</i> parameter refers to a class that is not already included in the Application installation. For example, if your application server is JBoss but you need to communicate with an external Weblogic JMS server, you need to install the jar file that includes the <i>weblogic.jndi.WLInitialContextFactory</i> class. You can obtain the necessary jar file from the corresponding vendor or your trading partner.
Platform availability	All supported Application platforms
Related services	JMS Queue adapter
Application requirements	No
Initiates business processes?	Initiates a business process when configured for async receive.
Invocation	This adapter can only be used in a business process when configured for sending or sync receive.

How the JMS Topic Adapter Works

The JMS Topic adapter is a *stateful* adapter; therefore, once the adapter is started, it establishes and maintains the connection to the configured Topic. The adapter can be configured to work in one of three modes: send, sync receive, or async receive.

Send Mode

When configured for Send mode, the adapter waits to be invoked by a business process. The adapter can either send a single workflow document in one invocation or it can send multiple workflow documents in one invocation (batch mode). Each workflow document is sent as a separate message. See *Invoking Batch Sending* on page 790.

If connection to the JMS Server is lost, JMS Topic adapter attempts to reestablish connection with the JMS Server with a retry delay of 60 seconds (60000 milliseconds) between two attempts. JMS Topic adapter attempts a maximum of twenty times to reestablish connection with the JMS Server.

Sync Receive

When configured for Sync Receive mode, the adapter waits to be invoked by a business process. Unlike during Async Receive mode, messages remain on the server until this adapter is invoked to receive the data. One advantage of using Sync Receive mode is that multiple messages can be received in one invocation of the adapter (batch mode). The number of messages received in one invocation can be limited, if necessary. Each message received is placed into the current workflow as a separate document. See *Invoking Batch Receiving* on page 791.

Async Receive

When configured for Async Receive mode, the adapter cannot be invoked by a business process. When the adapter starts and the session is established it registers an asynchronous callback listener so that messages are received when they become available. Additionally, a new workflow is started (bootstrapped) to process each message. See *Invoking Batch Receiving* on page 791.

Implementing the JMS Topic Adapter

To implement the JMS Topic adapter, complete the following tasks:

1. Activate your license for the JMS Topic adapter.
2. Set up a topic in your JMS server.
3. Create a JMS Topic adapter configuration. See *Creating a Service Configuration*.
4. Configure the JMS Topic adapter. See *Configuring the JMS Topic Adapter* on page 785.
5. Create a business process that includes the JMS Topic adapter and enable it.
6. Test the business process and the adapter.
7. Run the business process.

Configuring the JMS Topic Adapter

To configure the JMS Topic adapter, you must specify field settings in Application and in the Graphical Process Modeler (GPM).

Application Configuration

The following table describes the fields used to configure the JMS Topic adapter in Application:

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
Initial Context Factory (InitJndiFactory)	Initial context factory for connecting to the remote JMS Topic server. Used for JNDI lookup. For example, weblogic.jndi.WLInitialContextFactory. Required.
URL (JndiUrl)	Uniform Resource Locator of the application server that listens for connection requests. Required.
Remote Topic name (RemoteQueueTopicName)	Name of the remote JMS Topic that you want to exchange messages with. Required.
Remote Topic Connection Factory (Factory)	Encapsulates connection configuration information and enables JMS applications to create a connection with predefined attributes. Defines and configures one or more connection factories, and the JMS server adds them to the JNDI space during startup. The default is javax.jms.TopicConnectionFactory. Required.
Remote User Name (Username)	User name for accessing the JMS Server. Required if the JMS Server requires security credentials.
Remote Password (Password)	Password for accessing the JMS Server. Required if the JMS Server requires security credentials.
Turn on debug messages (Debug)	<p>Whether to log debug messages for this adapter instance. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – Debug messages will be logged. ◆ No – Debug messages will not be logged.
Topic Type (Action)	<p>Type of topic to access. Valid values are:</p> <ul style="list-style-type: none"> ◆ Topic Send – Sends messages. ◆ Topic Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message. ◆ Topic Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Field	Description
Message Type (Payload)	Type of message to send. Used only if topic type is Topic Send. Valid values are: <ul style="list-style-type: none"> ◆ ByteMessage ◆ ObjectMessage ◆ StreamMessage ◆ TextMessage
Bootstrap Workflow (InitialWorkFlowId)	Business process to initiate when data is received. Used only if topic type is Topic Receive Async. Required.
Document Storage Type (docStorageType)	Defines how the document will be stored in the system. Used only if topic type is Topic Receive Async. Required. Valid values: <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Note: See <i>Selecting a Document Storage Method for Bootstrap Adapters</i>.</p>
Maximum Bootstrap Threads (MaxThreads)	Maximum number of threads used when receiving files and starting business processes. Used only if topic type is Topic Receive Async. Each message received uses one thread. Default is 10. Optional.
Buffer Size (BufferSize)	Size of the buffer when receiving data. Used only if topic type is Topic Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional.
Document Filename (OutputFileName)	If you choose Topic Receive Async as the topic type for the JMS Topic adapter, then you can specify a file name for the data that the JMS Topic receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form <nodename>_yyyymmddhhmmsslll.
Connection retry attempts (RetryCount)	Maximum number of connection retry attempts. Used only if topic type is Topic Receive Async. Specify -1 for an infinite number of retry attempts. Default is 20. Optional.
Delay between retries (RetrySleep)	Number of milliseconds to wait between retry attempts. Default is 300000 ms (5 minutes). Used only if topic type is Topic Receive Async. Optional.
Notification Workflow (NotifyWorkFlow)	Business process initiated by the JMS Topic adapter if the maximum number of connection retries specified in <i>Connection retry attempts</i> is exceeded. Used only if topic type is Topic Receive Async. Required. If the adapter does not initiate a business process, select Not Applicable.

Field	Description
User	User ID to use for running the adapter. Select a user ID from the list. Valid values: Any valid Application user ID Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.
Jar Locations	Optional. Specify the preferred libraries of the jar files to be loaded with the JMS Topic adapter. You must specify the full path of the location of the jar files. Use semicolon (;) to separate multiple paths.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the JMS Topic service:

Parameter	Description
batchSndFilter	Optional. Only used when sending. If specified in the business process, triggers batch mode sending based on the documents that match the filter. You can use an asterisk '*' in the filter as a wildcard.
batchRcvLimit	Optional. Only used when receiving synchronously. If specified in the business process, the number of messages received is limited to the number specified. If not specified, all messages available are received.
batchRcvTimeout	Optional. Only used when receiving synchronously. If specified in the business process, it overrides the default receive timeout. If not specified, the default timeout is 2000 milliseconds (2 seconds).

Setting JMS Header Object Properties

When sending, you can set JMS object properties within the JMS header that are not part of the payload data. You can specify name/value pairs during runtime within the BPML. Because the user defined name/value pairs are unknown ahead of time, they cannot be set in the Application or GPM configuration so they must be manually added directly in the BPML. The JMS Topic adapter will look in ProcessData for the XML node name JMSetProperty and use any child nodes it finds to set the name/value pairs. There is a list of reserved property names that will set specific JMS message properties. An example of the ProcessData XML tree would look like this:

```
<ProcessData>
  <JMSetProperty>
    <somename1>somevalue1</somename1>
    <somename2>somevalue2</somename2>
  </JMSetProperty>
  <Reserved names that set specific JMS message properties>
    <correlationID>someStringValue</correlationID >
    <deliveryMode>someIntegerValue</deliveryMode>
```

```

    <destination>someTopicName</destination>
    <expiration>someLongValue</expiration>
    <messageID>someStringValue</messageID>
    <priority>someIntegerValue</priority>
    <redelivered>someBooleanValue(true/false)</redelivered>
    <replyTo>someTopicName</replyTo>
    <timestamp>someLongValue</timestamp>
    <type>someStringValue</type>
  </JMSetProperty>
</ProcessData>

```

An example of BPML that could be used to set these ProcessData name/value pairs follows:

```

<assign to="JMSetProperty/somename1" from="'somevalue1'" append="true"/>
<assign to="JMSetProperty/somename2" from="'somevalue2'" append="true"/>

```

When receiving, the JMS Topic adapter will set ProcessData items for all the JMS header fields and any object properties. Any object properties set in the JMS header will be put into ProcessData with the node name of JMS. For example, if there is a property called *somename* with a value of *somevalue*, ProcessData will contain JMS/somename with the corresponding value:

```

<JMS>
  <somename>somevalue</somename>
</JMS>

```

In addition to the user defined properties, the JMS Topic adapter will also set the following JMS header fields in ProcessData (if they are not null):

- JMS/correlationID
- JMS/deliveryMode
- JMS/destination
- JMS/expiration
- JMS/messageID
- JMS/priority
- JMS/redelivered
- JMS/replyTo
- JMS/timestamp
- JMS/type

The JMSetProperty can be used as a global property (under the ProcessData node) or a local property (under individual documents). Local JMSetProperty parameters override any global parameters and are useful when sending in batch mode. In the the below example, the global JMSetProperty has a parameter called “test” with a value of zero. Since the PrimaryDocument does not have a local JMSetProperty, it uses the global one. However, since doc1, doc2, and doc3 have local JMSetProperty parameters, they use the local parameters.

```

<ProcessData>

```

```

<JMSetProperty>
  <test>0</test>
</JMSetProperty>
<PrimaryDocument SCIObjectID="1833955:1063b363ed5:-774a" />
<doc1 SCIObjectID="1833955:1063b363ed5:-774b">
  <JMSetProperty>
    <test>1</test>
  </JMSetProperty>
</doc1>
<doc2 SCIObjectID="1833955:1063b363ed5:-774c">
  <JMSetProperty>
    <test>2</test>
  </JMSetProperty>
</doc2>
<doc3 SCIObjectID="1833955:1063b363ed5:-774d">
  <JMSetProperty>
    <test>3</test>
  </JMSetProperty>
</doc3>
</ProcessData>

```

Invoking Batch Sending

If a business process contains multiple documents in ProcessData, the JMS adapter can be invoked once with the workflow parameter **batchSndFilter**, which enables the adapter to send multiple messages for each of the documents that match the batchSndFilter criteria.

To invoke batch sending:

1. You do not need to make changes to the main adapter configuration; just add the appropriate assignment to the business process in the JMS adapter invocation step.

An example ProcessData for the example BPMLs below would look like this:

```

<ProcessData>
  <PrimaryDocument SCIObjectID="fe64b9:1060cac437b:-6a2a" />
  <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2b" />
  <XYZ>
    <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2c" />
    <doc2 SCIObjectID="fe64b9:1060cac437b:-6a2d" />
    <doc3 SCIObjectID="fe64b9:1060cac437b:-6a2e" />
  </XYZ>
</ProcessData>

```

Example 1

Sends all documents in ProcessData (including the PrimaryDocument). In this example, all five documents in ProcessData above are sent.

```

<operation name="JMS batch send">
  <participant name="JMSadapter" />
  <output message="toService">
    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'*'" />
  </output>
  <input message="fromService">

```

```
    <assign to="." from="*" />
  </input>
</operation>
```

Example 2

Sends all documents that begin with “doc” under the XYZ node. In this example, only three documents in the ProcessData above are sent.

```
<operation name="JMS batch send">
  <participant name="JMSadapter" />
  <output message="toService">
    <assign to="." from="*" />
    <assign to="batchSndFilter" from="'XYZ/doc*'"/>
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>
```

Invoking Batch Receiving

The type of receive adapter you choose to use is based on your business needs. If you are processing a large volume of messages, you may find that batching them is more efficient than bootstrapping one workflow for every message.

For the JMS Topic adapter, there are two types of receive queues:

Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message.

Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Additionally, there are two business process parameters associated with Sync Receive (batch receive):

batchRcvLimit – Optional—if used, this parameter limits the number of messages batched into the bootstrapped workflow. Default = no limit.

batchRcvTimeout – Optional—if used, this parameter specifies how long the adapter waits without receiving a message before ending. Default = 2000 (milliseconds)

Once a Sync Receive adapter completes the receive process, it creates the following information in ProcessData for the current workflow that invoked the adapter:

JMS/DocumentCount – This parameter is always created to show how many documents were created from messages received, even if zero messages were received.

JMS/Documentxxx – For every message received, a document is created under the JMS node and then sequentially numbered starting with one (that is, Document1, Document2, and so forth).

Another difference between Async Receive mode and Sync Receive mode is where the message metadata is stored in ProcessData. In Async Receive mode, it only creates one document (the PrimaryDocument), so

all the metadata is stored as JMS/<metadataName>. However, in Sync Receive mode, the metadata is stored under each document as JMS/<documentName>/<metadataName> as shown in the example below.

Example of ProcessData after a batch receive was performed:

```
<ProcessData>
  <JMS>
    <DocumentCount>3</DocumentCount>
    <Document1 SCIObjectID="1833955:1060de6d03d:-697b">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156360801</messageID>
      <priority>4</priority>
      <timestamp>1125515636080</timestamp>
    </Document1>
    <Document2 SCIObjectID="1833955:1060de6d03d:-6978">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361102</messageID>
      <priority>4</priority>
      <timestamp>1125515636110</timestamp>
    </Document2>
    <Document3 SCIObjectID="1833955:1060de6d03d:-6975">
      <redelivered>>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361243</messageID>
      <priority>4</priority>
      <timestamp>1125515636124</timestamp>
    </Document3>
  </JMS>
</ProcessData>
```

Lightweight Directory Access Protocol (LDAP) Adapter

The following table provides an overview of the LDAP adapter:

System name	None
Graphical Process Modeler (GPM) category	All Services
Description	Enables Application to communicate with local or remote LDAP servers using a Java Naming Directory Interface (JNDI). You can operate on data entries contained on an LDAP server, but you cannot change the underlying structure of those entries. The LDAP adapter supports Create, Read, Update, and Delete (CRUD) operations.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Restrictions	<p>The LDAP adapter supports LDAP versions 2 and 3 to the extent that the Sun LDAP/JNDI libraries do, except as noted in the following points:</p> <ul style="list-style-type: none">◆ The following standard LDAP operations are not supported:<ul style="list-style-type: none">- Adding new entry types- Adding or removing attributes from an entry- Referrals◆ Multi-value fields are supported for read, create, and update operations, but all values are treated as a single replacement value in the update operation; that is, the LDAP adapter cannot support updates of just one value in a multi-value field.◆ The LDAP adapter supports only the simple authentication type.◆ To prevent usernames and passwords from being accessible to other system users, enter usernames and passwords as instance variables only, not the BPML itself.

Requirements

To use the LDAP adapter, you must meet the following requirements:

Knowledge Requirements

To set up and use the LDAP adapter, you should know how to:

- Use the LDAP data model
- Use the Map Editor and the Translation service
- Apply XML concepts

Supported LDAP Versions

The LDAP adapter supports LDAP versions 2 and 3 to the extent that the Sun LDAP/JNDI libraries do, except as noted in the following points:

The following standard LDAP operations are not supported:

- ◆ Adding new entry types
- ◆ Adding or removing attributes from an entry
- ◆ Referrals

Multi-value fields are supported for read, create, and update operations, but all values are treated as a single replacement value in the update operation; that is, the LDAP adapter cannot support updates of just one value in a multi-value field.

The LDAP adapter supports only the simple authentication type.

System Requirements

For the LDAP adapter to work correctly, verify that:

- You have a valid logon ID and password and can access the remote LDAP server
- You can make a physical connection to the LDAP server from Application at run time

How the LDAP Adapter Works

Using LDAP directories is a popular method for storing and retrieving simple data in a hierarchical structure. LDAP works well with data on a wide area network (WAN).

Note: The LDAP adapter is not used in the authentication of external users of Application. This process does use an LDAP server, but not the LDAP adapter.

Java Naming Directory Interface (JNDI)

LDAP servers organize data into a hierarchical structure. An LDAP directory enables you to search a structured data repository and is optimized for read operations, unlike databases. Each record in the informational hierarchy can contain one or more fields or attributes. Each attribute can contain one or more values.

LDAP servers are not databases, although they can use databases to implement data storage. This distinction is important because LDAP may not support many sophisticated database features, such as advanced relational queries with table joins and transactional integrity across multiple operations. At present, the adapter accesses data in an LDAP server through the JNDI/LDAP API. The JNDI/LDAP API enables selection of LDAP data elements by name.

In general terms, LDAP is an example of a schema-based Operational Support System (OSS) as opposed to a service-based OSS. To make LDAP more service-based, Application overlays a standard service layer named CRUD (Create, Read, Update, and Delete) to manipulate data. The service layer works as follows:

Create – Adds a new entry to a directory and provides data for any attribute that already exists in the entry.

Read – Provides an entry search filter; the retrieved data is in Directory Service Markup Language (DSML) format.

Update – Modifies an LDAP entry. You need to provide a base distinguished name (baseDN) to identify the entry, and the names and values of the attributes to update.

Delete – Deletes an entry from the LDAP server.

Accessing Data

To access data, the LDAP adapter needs these items:

Service to perform – Create, Read, Update, Delete

A unique name that specifies a record on the LDAP server

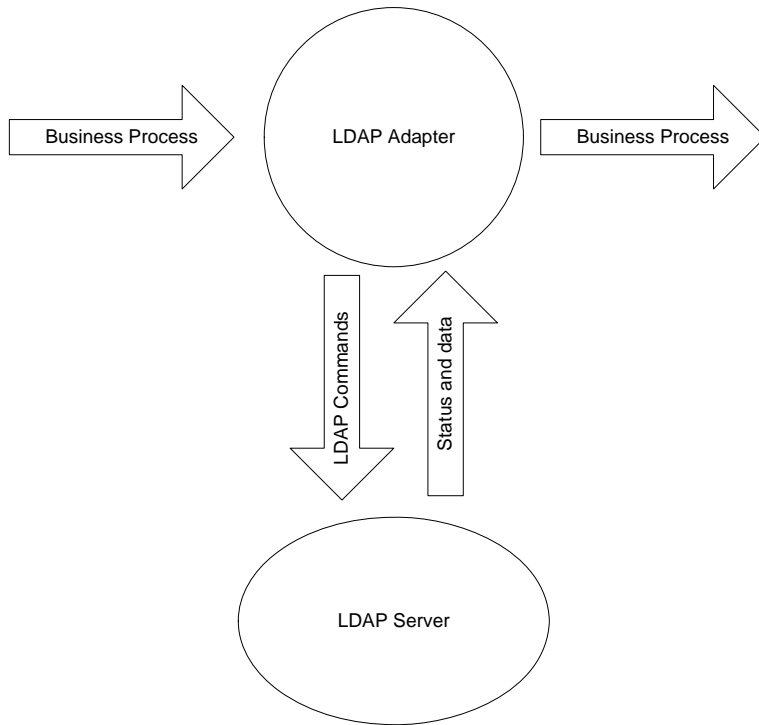
Field names within the record

Adapter Process

The following steps summarize how the LDAP adapter works:

1. The LDAP adapter sends a request to the LDAP server.
2. The adapter takes the results returned from the LDAP server and places them back into the business process context.
3. The adapter passes the updated internal business process context back to the business process.
4. The adapter is ready to process the next request.

The following figure shows how the LDAP adapter communicates with the LDAP server within a business process:



The following steps summarize how the LDAP adapter communicates with the LDAP server within a business process:

1. The Translation service checks the translation object (.txo) in to Application for later use by the Translation service.
2. When initiating the business process, the user supplies the name and location of the customer input document that has the necessary information to be retrieved, such as the customer name and the name of fields.
3. From the business process, the business process engine (BPE) receives the name of the Translation object (identified from the list of maps checked in to Application).
4. When the business process is started, the BPE starts the Translation service. The Translation service builds the XML file for the LDAP adapter with the necessary information from the customer document.
5. The LDAP adapter uses the information in the file to make a connection with the LDAP server and retrieve the required information by making appropriate calls.
6. The LDAP adapter constructs an XML document with the retrieved data.
7. The XML document is passed to the business process.
8. Application performs the next step in the business process.

For example, consider the following scenario. You have customer information stored in an LDAP server (which has an internal database). To provide the sales department with the customer contact information from the LDAP database, use the LDAP adapter to access this information and then write the information to disk using the File System adapter.

Implementing the LDAP Adapter

To implement the LDAP adapter:

1. Create an LDAP adapter service configuration. For information, see *Managing Services and Adapters*.
2. Configure the LDAP adapter. For information, see *Configuring the LDAP Adapter* on page 797.
3. Create XML documents, as necessary. For information, see *Creating XML Documents for the LDAP Adapter* on page 798.
4. Use the LDAP adapter in a business process.

Configuring the LDAP Adapter

The following tables describe fields used to configure the LDAP adapter in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Host Name (hostName)	Name or IP address of the host running the LDAP server. Can be overridden in a business process or process data.
Port (port)	IP port number on the host. Default is 389. Can be overridden in a business process or process data.
Read Timeout in Secs (readTimeOut)	Timeout value in seconds. Default is No limit (0).
Max Number of Records to Read (maxReadRecords)	Maximum number of records to return from the LDAP server. Zero (0) means no limit is applied.
Set Authentication? (LDAPAuthentication)	Whether you will attempt to connect to the LDAP server with authentication, or anonymously.
Login Name (loginName)	Login name for the host LDAP server. Can be overridden in a business process or process data.
Password (password)	LDAP server password for the associated login name. Can be overridden in a business process or process data.

Creating XML Documents for the LDAP Adapter

For LDAP adapter business processes, the XML document passed into the adapter determines the operation to start on the LDAP server. The XML document must correspond to one of four Document Type Definitions (DTDs), which define the CRUD operations (Create, Read, Update, or Delete). In other words, the DTDs verify that the XML is correct for the operation it is written for.

The LDAP adapter provides the DTDs that define the XML that is passed to and received from the adapter. The Map Editor and the Translation service use these DTDs to verify data conversions to and from customer formats.

To check out the DTDs from Application and load them to your local disk:

1. From the **Deployment** menu, select **Schemas**.
2. In the XML Schemas window, search for the LDAP DTDs.
They are named LDAPCreate.dtd, LDAPRead.dtd, LDAPUpdate.dtd, LDAPDelete.dtd, and dsml.dtd.
3. Click **Source Manager**.
4. Save each DTD to your local disk.

XML Construction

When constructing XML for the LDAP adapter, remember the following points:

In the request element, the operation attribute specifies the operation to be performed (Create, Read, Update, or Delete), and:

- ◆ The Base Distinguished Name (baseDN) attribute is different for each operation.
- ◆ The Scope attribute specifies search scope and is used only in Read operations.

Parameter elements identify fields, and:

- ◆ Name attributes identify the field name.
- ◆ Usage attributes specify whether the data is input, output, or search.
- ◆ Type attributes specify the type of data to be sent in the output parameters (for example, text/none, bin/base64).

Note: Usage attributes other than those specified for a particular operation are discarded. For example, if a request XML for a Create operation has input and output usage attributes, the input attribute is discarded.

Examples

The following XML excerpts are examples of input documents for Create, Read, Update, and Delete operations.

Create Operations

The baseDN attribute identifies the record to be created. The usage attribute is always *output* for Create operations.

Two values exist for Type attributes in Create operations:

text/none – Intended for ordinary textual data with no encoding. Default.

bin/base64 – For binary data. Encode as *base64* in the content of the param tag.

The following example shows a Create operation:

```
<LDAPAdapter>
  <request operation="Create" baseDN="uid=jblow, ou=People, o=isg.stercomm.com">
    <param.1 name="objectclass" type="text/none" usage="Output">top</param.1>
    <param.2 name="objectclass" type="text/none" usage="Output">person</param.2>
    <param.3 name="ou" type="text/none" usage="Output">People</param.3>
    <param.4 name="mail" type="text/none" usage="Output">jb@ab.com</param.4>
    <param.5 name="uid" type="text/none" usage="Output">jblow</param.5>
    <param.6 name="sn" type="text/none" usage="Output">Blow</param.6>
    <param.7 name="givenname" type="text/none" usage="Output">Joe</param.7>
  </request>
</LDAPAdapter>
```

Read Operations

The baseDN attribute identifies the highest point in the hierarchy to begin the search, and the Scope attribute defines the extent of the search. The Scope attribute values are:

subTree

base

oneLevel

The search filter is a query string and is denoted with a param element where the Usage attribute is *search*. Use the following symbols to define the search:

Symbol	Represents
Parentheses ()	Enclosed group of compares
Ampersand &	Logical operator AND
Pipe	Logical operator OR
Exclamation point !	Logical operator NOT

The logical operator should appear before the parentheses enclosing the group of compares that the logical operator affects. For example:

```
(&(cn=X)(sn=Y))
```

This example means that cn is equal to X and sn is equal to Y.

One param element must have a usage attribute value of *search* and must contain the search specification as defined by the LDAP model. The rest of the param elements, if any, specify the field names to be retrieved from the LDAP server for the records that match the search filter.

The following example shows a retrieval for the cn field and the jpegphoto field:

```
<LDAP Adapter>
  <request scope="subtree" operation="Read" baseDN="uid=jblow, ou=People,
o=isg.stercomm.com">
    <param.1 usage="Search">(&(objectclass=person)(sn=Blow))</param.1>
    <param.2 name="jpegphoto" usage="Input"/>
  </request>
```

```
<param.3 name="cn" usage="Input" />
</request>
</LDAP Adapter>
```

Read Output Documents

Because Read is the only operation that has data returned, an output document is associated with this operation. The output document is written in DSML and is added to the business process context when the operation successfully completes. DSML is a standard representation of directory information in XML format. The LDAP adapter is compatible only with DSML Version 1.0.

DSML is intended to be a simple XML schema definition that enables directories to publish basic profile information. Find the full specifications for DSML at www.dsml.org.

The following example shows DSML for a Read output document:

```
<dsml>
  <directory-entries>
    <entry dn="uid=scarte2, ou=People, o=isg.stercomm.com">
      <attr name="telephonenumber">
        <value>+1 408 555 6022</value>
      </attr>
      <attr name="mail">
        <value>scarte2@isg.stercomm.com</value>
      </attr>
      <attr name="uid">
        <value>scarte2</value>
      </attr>
    </entry>
  </directory-entries>
</dsml>
```

Update Operations

The baseDN attribute identifies the record to be updated. Two values exist for Type attributes in Update operations:

text/none – Intended for ordinary textual data with no encoding. Default.

bin/base64 – Intended for binary data. Encode as *base64* in the content of the param element.

The parameter element usage attribute value is always output.

For multivalue fields, multiple param elements can have the same name, but the content is different for each. The following example shows a multivalue *first name* field:

```
<LDAPAdapter>
  <request operation="update" baseDN="uid=jblow, ou=People, o=isg.stercomm.com">
    <param.1 name="employeenumber" type="text/none" usage="Output">1234</param.1>
    <param.2 name="firstname" type="text/none" usage="Output">Joe</param.2>
    <param.3 name="firstname" type="text/none" usage="Output">Joseph</param.3>
    <param.4 name="firstname" type="text/none" usage="Output">Joey</param.4>
  </request>
</LDAPAdapter>
```

Delete Operations

The baseDN attribute identifies the record to be deleted. The following example shows a Delete operation:

```
<LDAPAdapter>
```



```
<request operation="Delete" baseDN="uid=jblow, ou=People, o=isg.stercomm.com">
  </request>
</LDAPAdapter>
```

Lightweight Java Database Connectivity (JDBC) Adapter (Build 4300 - Build 4320)

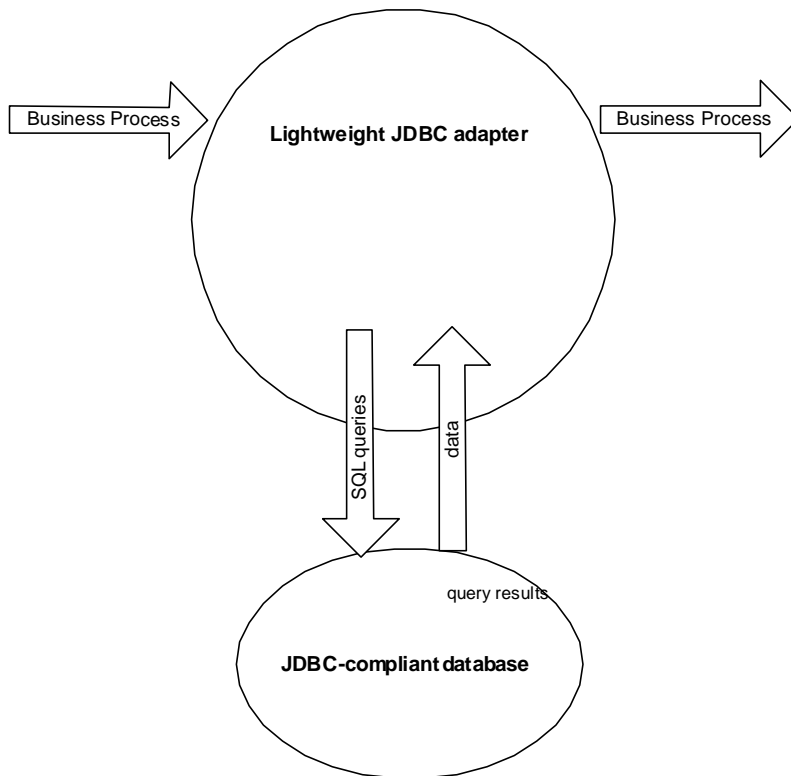
The following table provides a high-level overview of the Lightweight JDBC adapter:

System name	None
Graphical Process Modeler (GPM) category	All services, Translation
Description	<p>Enables you to retrieve data from a JDBC-compliant database or update a JDBC-compliant database as part of a business process within the Application, using one of the following methods:</p> <ul style="list-style-type: none">◆ Specify XPath queries to retrieve information from process data and set the Lightweight JDBC (LWJDBC) adapter parameters in a business process. Use XPath queries to set the Lightweight JDBC adapter parameters in a business process, allowing for reuse of the same instance of a Lightweight JDBC adapter configuration multiple times throughout a business process. To use XPath queries, you need to first load into process data the data which you want to use to set the Lightweight JDBC adapter parameters. You can use either the XML Encoder service or DocToDOM function in your business process to load into process data a primary document containing the data you want to use for XPath queries.◆ Specify constants in the Lightweight JDBC adapter configuration. Unlike the JDBC adapter, the Lightweight JDBC adapter does not use a map in which the SQL statements are fixed. The Lightweight JDBC adapter enables you to specify parameters in the SQL statement that are supplied at run time. For information about the JDBC adapter, see <i>Java Database Connectivity (JDBC) Adapter</i>. The Lightweight JDBC adapter executes a query, and you receive an XML document that contains the results. This document becomes the primary document in the business process. Then you can either:<ul style="list-style-type: none">◆ Load the XML document into process data.◆ Carry the XML document forward into another service in the business process.
Business usage	<p>Use the Lightweight JDBC adapter if you want to retrieve or manipulate data and need the flexibility of specifying various SQL queries.</p> <p>Use the JDBC adapter if you have complex output and you want to manipulate the data from a database and control the structure of the output.</p>
Preconfigured?	No
Requires third party files?	You must configure a connection to an external database for the Lightweight JDBC adapter.
Platform availability	All supported Application platforms
Related services	None
Initiates business processes?	Yes
Invocation	Runs by a schedule or business process.

How the Lightweight JDBC Adapter Works

The Lightweight JDBC adapter can start a business process or it can be used in a business process. This service can be scheduled to run at weekly or timed intervals.

The following figure shows how the Lightweight JDBC adapter communicates with a JDBC-compliant database:



Business Scenario

Your company receives information from an external database about a customer that needs to be either added (if the customer does not exist) or updated in your internal customer database. The customer information arrives in XML format.

Business Solution Example

The following approach is used to solve the business scenario:

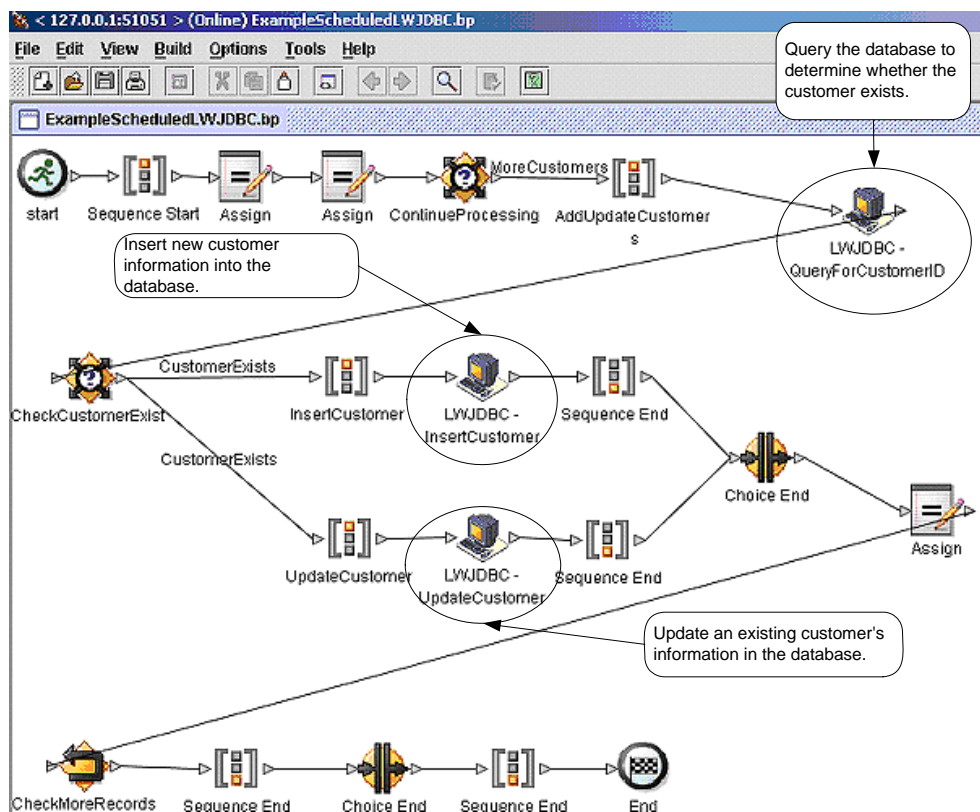
1. Configure a Lightweight JDBC adapter instance to be included in a business process. This one instance of the Lightweight JDBC adapter is used multiple times throughout the example business process to perform the following tasks:
 - ◆ Query the database for the customer ID passed into the business process to determine if the customer currently exists in the internal customer database.
 - ◆ Add the customer information to the customer database if the customer does not exist.
 - ◆ Update the customer information in the customer database if the customer does exist.

2. Create a business process that determines whether to add or update the customer information to the internal customer database. The Lightweight JDBC adapter instance configured above is used in this business process to perform the query, add, and/or update.
3. Configure a second Lightweight JDBC adapter instance to retrieve customer records from the external database and to start the business process created in the previous step. This Lightweight JDBC adapter instance is also scheduled to run at 10 p.m. daily to retrieve customer records from the external database and return the results to the business process.

GPM Example

The following example shows a solution to the business scenario using the GPM.

A Lightweight JDBC adapter has been configured and scheduled to retrieve records from an external database. The result from the query is the primary document to this business process.



The Assign statements are used to retrieve the customer information from the primary document and place the information into process data. The parameters specified for the Lightweight JDBC adapter, which includes customer information retrieved from process data and SQL statements, are input to the adapters. A primary document containing the results from the SQL statements issued against the database is output from the Lightweight JDBC adapter instances and is passed to the next step in the business process.

For information about the Lightweight JDBC parameters, see *Implementing the Lightweight JDBC Adapter* on page 806.

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML.

```
<process name="ExampleScheduledLWJDBC">
<rule name="MoreCustomers">
  <condition>NumberOfRecords &gt; 0</condition>
</rule>
<rule name="CustomerExists">
  <condition>number(/ProcessData/NumberOfCustomers) = 0 </condition>
</rule>
<sequence>
  <assign to="NumberOfRecords" from="number(count(DocToDOM(PrimaryDocument)/Customer/
customer_id))" append="true"/>
  <assign to="CustomerInformation" from="DocToDOM(PrimaryDocument)" append="true"/>
  <choice name="ContinueProcessing">
    <select>
      <case ref="MoreCustomers" activity="AddUpdateCustomers"/>
    </select>
    <sequence name="AddUpdateCustomers">
      <operation name="LWJDBC - QueryForCustomerID">
        <participant name="ExampleLWJDBCBusinessProcess"/>
        <output message="LightweightJDBCAdapterTypeInputMessage">
          <assign to="param1" from="number(/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_id/text())"/>
          <assign to="paramtype1">Integer</assign>
          <assign to="query_type">SELECT</assign>
          <assign to="result_name">RootResultXmlElement</assign>
          <assign to="row_name">ResultsFromSQL</assign>
          <assign to="sql">SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2 WHERE CUSTOMER_ID = ?</
assign>
          <assign to="." from="*" />
        </output>
        <input message="inmsg">
          <assign to="NumberOfCustomers" from="DocToDOM(PrimaryDocument)/ResultsFromSQL/
CUSTOMERS" append="true"/>
        </input>
      </operation>
      <choice name="CheckCustomerExist">
        <select>
          <case ref="CustomerExists" activity="InsertCustomer"/>
          <case ref="CustomerExists" negative="true" activity="UpdateCustomer"/>
        </select>
        <sequence name="InsertCustomer">
          <operation name="LWJDBC - InsertCustomer">
            <participant name="ExampleLWJDBCBusinessProcess"/>
            <output message="LightweightJDBCAdapterTypeInputMessage">
              <assign to="param1" from="number(/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_id/text())"/>
              <assign to="param2" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_name/text()"/>
              <assign to="param3" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_address/text()"/>
              <assign to="param4" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_phone/text()"/>
            </output message>
          </operation>
        </sequence>
      </choice>
    </sequence>
  </choice>
</sequence>
</process>
```

The LWJDBC instance used to determine if the customer exists

The LWJDBC instance used to add a customer to the database

```

    <assign to="paramtype1">Integer</assign>
    <assign to="paramtype2">String</assign>
    <assign to="paramtype3">String</assign>
    <assign to="paramtype4">String</assign>
    <assign to="query_type">ACTION</assign>
    <assign to="result_name">RootResultXmlElement</assign>
    <assign to="row_name">ResultsFromInsert</assign>
    <assign to="sql">INSERT INTO CUSTOMER2 VALUES(?, ?, ?, ?)</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="." />
  </input>
</operation>
</sequence>
<sequence name="UpdateCustomer">
  <operation name="LWJDBC - UpdateCustomer">
    <participant name="ExampleLWJDBCBusinessProcess" />
    <output message="LightweightJDBCAdapterTypeInputMessage">
      <assign to="param1" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_name/text()" />
      <assign to="param2" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_address/text()" />
      <assign to="param3" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_phone/text()" />
      <assign to="param4" from="number(/ProcessData/CustomerInformation/ResultsFromQuery.
Customer[number(//NumberOfRecords)]/customer_id/text())" />
      <assign to="paramtype1">String</assign>
      <assign to="paramtype2">String</assign>
      <assign to="paramtype3">String</assign>
      <assign to="paramtype4">Integer</assign>
      <assign to="query_type">ACTION</assign>
      <assign to="result_name">RootResultXmlElement</assign>
      <assign to="row_name">ResultsFromUpdate</assign>
      <assign to="sql">UPDATE CUSTOMER2 SET CUSTOMER_NAME = ?, CUSTOMER_ADDRESS=?,
CUSTOMER_PHONE=? where CUSTOMER_ID=?</assign>
      <assign to="." from="*" />
    </output>
    <input message="inmsg">
      <assign to="." from="." />
    </input>
  </operation>
</sequence>
</choice>
  <assign to="NumberOfRecords" from="NumberOfRecords - 1" />
  <repeat name="CheckMoreRecords" ref="ContinueProcessing" />
</sequence>
</choice>
</sequence>
</process>

```

The LWJDBC instance used to update an existing customer's information in the database

For information about the Lightweight JDBC parameters, see *Implementing the Lightweight JDBC Adapter* on page 806. For additional Lightweight JDBC adapter examples, including examples for using Oracle stored procedures and functions, see *Lightweight JDBC Business Process Usage* on page 827.

Implementing the Lightweight JDBC Adapter

You can implement the Lightweight JDBC adapter by:

Configuring a Lightweight JDBC adapter instance to include in business processes. This lets you reuse the same Lightweight JDBC adapter instance to run different SQL queries against a database.

Configuring a Lightweight JDBC instance that runs an SQL query and then starts a business process.

The information in this section applies to both implementation types.

To implement the Lightweight JDBC adapter, complete the following tasks:

1. Collect the following information:
 - ◆ The name of the business process (if the adapter is to start a business process)
 - ◆ Database pool name used to connect to an external database
 - ◆ Parameter values for the `jdbc_customer.properties.in` file (see *Adding New Database Pools* on page 807).
2. Set up a connection to an external database, if you have not already done so. For information, see *Setting Up a Connection to an External Database* on page 807.
3. Create a Lightweight JDBC adapter configuration. For information, see *Managing Services and Adapters*.
4. Configure the Lightweight JDBC adapter. For information, see *Configuring the Lightweight JDBC Adapter* on page 817.
5. Use the Lightweight JDBC adapter in a business process.

Note: If the Lightweight JDBC adapter configuration starts a business process, create the business process prior to configuring the adapter.

Setting Up a Connection to an External Database

You must set up a connection to an external database for the Lightweight JDBC adapter. You can use any of the databases supported by the Application for internal use or other JDBC-compliant databases, such as Sybase. For information about databases supported by the Application, see *System Requirements*.

Adding New Database Pools

To define a new database pool for use by the Lightweight JDBC adapter, you must add settings for the pool to the `jdbc_customer.properties.in` file, which is located in the Application `/install_dir/properties` directory.

In `jdbc_customer.properties.in`, specify the database server name, port number, database/catalog name, user ID and password. To encrypt your database password, use the `encrypt_string.sh` or `encrypt_string.cmd` utility in the `bin` directory. Then place the encrypted password, prefixed by an encryption indicator, in your properties file.

Caution: There are two `jdbc_customer.properties` files: `jdbc_customer.properties.in`, which is the “template” properties file; and `jdbc_customer.properties`, which is the “packaged” properties file.

It is extremely important to ensure that you add the records to the template file, `jdbc_customer.properties.in`, not to the packaged file.

Each time you run the `setupfiles` command in the Application, all packaged files are updated with

the information contained in their template (.in) files. This means that if you make changes to the packaged file, `jdbc_customer.properties`, they are lost each time `setupfiles` is run. Always make changes to the template file, `jdbc_customer.properties.in`, and your changes will be maintained.

If the database you want to connect to resides on a database server type that is not the same as the Application database server type, you also need to install a JDBC driver using the `install3rdparty.sh` or `install3rdparty.cmd` utility.

Select a table and column in your database to use in the test on reserve function. This function causes the Application to test the database connection using a quickly run query before attempting to use it. This function ensures that idle connections are revived. The column referenced in the query should be of the type *varchar* and should be at least five characters in length.

Connecting to an External Database

To connect to an external database:

1. Add the necessary records to the `jdbc_customer.properties.in` file found in the `/install_dir/properties` directory.

Note: If invalid data (like ABC or 13.45) is entered in a pool setting, the setting uses its default value.

See the examples that follow this procedure for *Oracle 8i/9i* on page 815, *DB2* on page 815, *MS SQL 2000* on page 816, and *Sybase* on page 817.

The following table contains the parameters needed to add a new database pool to the `jdbc_customer.properties.in` file:

Parameter	Description
<code>databasePool.driver</code>	JDBC driver class file for the database application.
<code>databasePool.url</code>	Database location (full URL as defined by the Java JDBC standards). Note: For Oracle systems, the last segment in the URL is the Oracle SID (not the System Reference or Tnsnames entry). Note: You can locate the Java JDBC standards on the java.sun.com Web site.
<code>databasePool.user</code>	Username for logging into the database.
<code>databasePool.password</code>	Password for logging into the database.
<code>databasePool.maxconn</code>	Maximum number of database connections for the connection pool.

Parameter	Description
<i>databasePool.storedProcClassName</i>	<p>Specifies the class that handles stored procedure calls for the Lightweight JDBC adapter.</p> <p>The following classes are used for the database types:</p> <ul style="list-style-type: none"> ◆ MSSQL, Sybase, and DB2 – com.sterlingcommerce.woodstock.util.frame.jdbc.GenericStoredProcQuery ◆ Oracle 8i/9i – com.sterlingcommerce.woodstock.util.frame.jdbc.OracleNoAppStoredProcQuery <p>Note: The Lightweight JDBC adapter does not support stored procedures for MySQL.</p>
<i>databasePool.varDataClassName</i>	<p>Each database that the Application supports handles binary objects differently. This parameter specifies the class used to handle binary data for the database. Enter the correct class for your database:</p> <ul style="list-style-type: none"> ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ISeriesVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2VarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ZOSVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.JConnectVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.MSSQLVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.MySQLVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.OracleBlobVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.OracleVarData
<i>databasePool.catalog</i>	Database name (usually the same as the last segment of the URL)
<i>databasePool.type</i>	<p>Valid values:</p> <ul style="list-style-type: none"> ◆ local ◆ remote (default)
<i>databasePool.testOnReserve</i>	<p>Whether to test the connection.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ true ◆ false <p>Note: This function causes the Application to test the database connection before attempting to use it and revives idle connections.</p>

Parameter	Description
<i>databasePool.testOnReserveQuery</i>	<p>SQL query to use when testing the connection. Select a table and column in your database to use in the test on reserve function. The column referenced in the query should be of the type <i>varchar</i> and should be at least five characters in length. This query needs to be executable by the <i>databasePool.username</i> account and must be a valid SQL query. For example:</p> <pre>SELECT table_name FROM user_tables WHERE table_name=?</pre> <p>where ? must accept a string value. The query does not have to return a value to operate. If the query fails, the Database Pool is not activated.</p>
<i>databasePool.testOnReserveInterval</i>	<p>The minimum number of milliseconds between running testOnReserve on the same connection. The default value is 60000. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 - No interval and current interval is used. ◆ <= 0 - No interval. ◆ > 0 - The minimum number of milliseconds between running testOnReserve on the same connection.
<i>databasePool.max8177RetryCount</i>	<p>Only used for an Oracle database, this tells the software how many times to retry if it receives an ORA-8177 error in certain situations.</p>
<i>databasePool.dbvendor</i>	<p>Enter the database name: sybase, oracle, mysql, mssql, db2, db2zos, db2iseries, or other vendor name.</p>
<i>databasePool.maxsize</i>	<p>Maximum size of the database pool. This property was previously contained in the poolManager.properties file. This value must not exceed the value specified for the databasePool.maxconn parameter in the jdbc.properties file.</p>
<i>databasePool.initsize</i>	<p>Initial size of the database pool. This property was previously contained in the poolManager.properties file.</p>
<i>databasePool.factory</i>	<p>Always enter the following: com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory</p>

Parameter	Description
<i>databasePool.behaviour</i>	<p>Behavior a connection pool exhibits when it runs out of connections. This property replaces the <i>databasePool.onEmpty</i> property in the former <i>poolManager.properties</i> file. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 – The pool simply returns indicating to the software to abort its current action and try again later. This value corresponds to the value return in the <i>databasePool.onEmpty</i> property. ◆ 1 – The pool waits the number of milliseconds specified in <i>databasePool.waittime</i> for a connection to be returned before indicating to the software to abort and try again. This value corresponds to the value <i>wait</i> in the <i>databasePool.onEmpty</i> property. ◆ 2 – The pool creates a buffered connection (a connection above the size specified in <i>databasePool.maxsize</i>). When using a setting of 2, the maximum number of connections for the pool is the value specified for <i>databasePool.maxsize</i> plus the value specified for <i>databasePool.buffersize</i> connections. This allows connections to be created under heavy demand. This value corresponds to the value <i>new</i> in the <i>databasePool.onEmpty</i> property.
<i>databasePool.lifespan</i>	<p>The number of milliseconds a connection will live in a given pool before it needs to be removed.</p> <ul style="list-style-type: none"> ◆ 0 - (Default) No timeout. ◆ <= 0 - No timeout. ◆ > 0 - Number of milliseconds that a connection stay in pool.
<i>databasePool.idletimeout</i>	<p>The number of milliseconds a connection can stay idle in a given pool before it needs to be removed. The default value is 86400000. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 - No timeout. ◆ <= 0 - No timeout. ◆ > 0 - Number of milliseconds that a connection stay in pool.
<i>databasePool.housekeepinginterval</i>	<p>The minimum number of milliseconds between running the housekeeping task to clean out idle connections. Valid values are any positive number. The default value is 3600000 milliseconds (1 hour). Any number less than 3600000 will cause the default of 3600000 milliseconds to be used.</p>
<i>databasePool.buffersize</i>	<p>Number of extra connections that the connection pool can create above the value specified for <i>databasePool.maxsize</i> to improve handling of unanticipated loads on the system. This property is only used if <i>databasePool.behavior</i> is set to 2.</p>
<i>databasePool.waittime</i>	<p>Amount of time (in milliseconds) to wait for a connection to become available before indicating to the software to abort the current action and try again later. This property is only used if <i>databasePool.behavior</i> is set to 1.</p>

2. Run the *setupfiles.sh* (UNIX) or *setupfiles.cmd* (Windows) utility located in the */install_dir/bin* directory of the Application installation directory. This updates the “packaged” properties file, *jdbc_customer.properties*, with the changes from the “template” properties file, *jdbc_customer.properties.in*.

3. If the vendor for the connection database is not the same vendor as the database vendor used for the Application database, install the appropriate JDBC driver to access the database server. Use the `install3rdParty.sh` (UNIX) or `install3rdparty.cmd` (Windows) utility located in the `/install_dir/bin` directory of the Application installation directory to add the JDBC driver jar file(s). Type **install3rdParty** on the command line to get a description of the parameters you can specify.

The following examples are for a UNIX environment. The vendor name and version are the first two parameters, along with the location of the zip file containing the JDBC driver files.

- ◆ For Oracle 9i, install the driver using the following command:

```
./install3rdParty.sh Oracle 9_2_0_5 -d  
/usr local directory/oracle/9_2_0_5/classes12.zip
```

- ◆ For DB2, install the driver using the following command:

```
./install3rdParty.sh db2java 7_2 -d / / / /db2java.zip
```

- ◆ To install the jConnect driver for Sybase, refer to *Installing a Sybase Driver* on page 812.

4. Stop and restart the Application to use the changed files.

Installing a Sybase Driver

Install the jConnect driver for Sybase using the following procedure:

1. Download `jConnect-5_5.zip` from the Sybase web site.
2. Run the following command:

```
./install3rdParty.sh jconnect 5_5 -d  
/ / / /5_5/jConnect-5_5.zip
```

- ◆ If this command succeeds, you are finished with this procedure.
- ◆ If Application reports in the system log that the driver could not be registered because the driver class cannot be found, continue with the procedure. Use the following steps to remove existing references to jConnect.

3. Stop Application.
4. Change your directory to `install_dir/jar`.
5. Delete any existing folders referencing jConnect.
6. Change your directory to `install_dir/properties`.
7. Open the files `dynamicclasspath.cfg` and `dynamicclasspath.cfg.in`. Delete any lines referencing jConnect, and save the files.
8. Create the following temporary directory:

```
install_dir/bin/jconnect
```

9. Extract only the jar files from `jConnect-5_5.zip` to this directory.

10. Run the following command:

```
install3rdparty jconnect 5_5 -d install_dir/bin/jconnect/*.jar
```

11. Check `install_dir/jar/jconnect/5_5/your_platform` to make sure that six jar files have been copied successfully.

12. Open the `dynamicclasspath.cfg` file in `install_dir/properties` and check that the following entries are there:

```
VENDOR_JAR=/          /jar/jconnect/5_5/your_platform
            install_dir          your_platform
            install_dir          your_platform/jisql.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2d.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/ribo.jar
```

13. Open the `customer.jdbc.properties.in` file in `install_dir/properties` and check that the following entries are there:

```
jconnectPool.driver=com.sybase.jdbc2.jdbc.SybDriver
jconnectPool.url=jdbc:sybase:Tds:your Hostname:4100/your DB
jconnectPool.user=your user name
jconnectPool.password=your password
jconnectPool.catalog=your database

jconnectPool.testOnReserveQuery=your Test On Reserve Query

jconnectPool.bufferSize=500
jconnectPool.maxSize=28
jconnectPool.initSize=1
jconnectPool.behaviour=2
jconnectPool.waitTime=1000
jconnectPool.storedProcClassName=
com.sterlingcommerce.woodstock.util.frame.jdbc.SybaseStoredProcQuery
jconnectPool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc
c.JConnectVarData
jconnectPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.Connect
ionFactory
```

14. Save the `customer.jdbc.properties.in` file and run the following command:

```
/bin/setupfiles.sh
```

This procedure should result in a successful connection to your Sybase database. However, if the database has been configured as character set `ROMAN8`, it is likely that you will see the following message in the Application system log, because of a limitation in the Sybase driver:

```
java.sql.SQLException: JZ0IB: The server's default charset of roman8 does not map
to an encoding that is available in the client Java environment. Because jConnect
will not be able to do client-side conversion, the connection is unusable and
is being closed. Try using a later Java version or try including your Java
installation's i18n.jar or charsets.jar file in the classpath
```

One resolution of this issue is to configure the primary Adaptive Server with a default character set that maps to one of the character sets supported by jConnect for JDBC (for example, UTF-8). For more details, refer to the Sybase documentation.

Another resolution of this issue is to use the open source jTDS driver from Sourceforge (sourceforge.net). To install this driver, follow these instructions:

1. Stop Application.
2. Remove references to jConnect as described previously.
3. Copy the jtds-1.2.jar file to an accessible directory on the Application machine.
4. Run the following command:

```
/bin/Install13rdparty.sh jTDS 1_2 - jar absolutePath
```

5. Check that the dynamicclasspath.cfg file has picked up this change. For example, */install_dir/jar/jTDS/1_2/your_platform/jtds-1.2.jar*.
6. Edit the jdbc_customer.properties.in file. The definition of the pool should be similar to the following example:

```
your user name  
your password  
your database
```

```
#jTDSPool.testOnReserveQuery=  
jTDSPool.dbvendor=jtds  
jTDSPool.bufferSize=50  
jTDSPool.maxSize=20  
jTDSPool.iniSize=5  
jTDSPool.behaviour=2  
jTDSPool.storedProcClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.  
.SybaseStoredProcQuery  
jTDSPool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.Ge  
nericVarData  
jTDSPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionF  
actory
```

7. Restart Application.

Properties File Examples for Specific Databases

Oracle 8i/9i

For Oracle 8i/9i, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool
databasePool      jdbc:oracle:thin:@servername:0000:servername
databasePool      username
databasePool      password
databasePool      catalogname
databasePool

                                SELECT TestConnection from Connection_tb WHERE
TestConnection = ?
databasePool
databasePool      n
databasePool
databasePool      n
databasePool      n
databasePool      n
databasePool      n
databasePool
databasePool
databasePool      n
databasePool
databasePool
databasePool
```

DB2

For DB2, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

Note: The JDBC adapter does not support stored procedures for DB2/iSeries and DB2/zOS.

```
database
database

database
database

database
database
database
database
```

database

database
database
database
database
database
database
database
database
database

database
database
database
database
database

MS SQL 2000

For MS SQL 2000, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool  
databasePool      jdbc:microsoft:sqlserver://servername:0000;DatabaseName=SQLdatabase  
;SelectMethod=cursor  
databasePool      username  
databasePool      password  
databasePool      catalogname  
databasePool  
databasePool      true  
databasePool      SELECT TestConnection from Connection_tb WHERE  
TestConnection = ?  
databasePool  
databasePool  
databasePool      n  
databasePool      n  
databasePool      n  
databasePool      n  
databasePool  
databasePool  
databasePool      n  
databasePool  
  
databasePool  
  
databasePool
```


Sybase

For Sybase, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool
databasePool      jdbc:sybase:Tds:servername:0000/SybaseDB
databasePool      username
databasePool      password
databasePool      catalogname
databasePool
databasePool      true
databasePool      SELECT TestConnection from Connection_tb WHERE
TestConnection = ?
databasePool
databasePool
databasePool      n
databasePool      n
databasePool      n
databasePool      n
databasePool
databasePool
databasePool
databasePool      n
databasePool
databasePool
```

databasePool

databasePool

Encrypting Your Database Password

To use encryption for the database password:

1. Use `encrypt_string.sh` (UNIX) or `encrypt_string.cmd` (Windows).
2. When prompted, enter your external database password.
The script returns the encrypted value for your password.
3. Place the encrypted password in your `jdbc_customer.properties.in` file entry (see step 2 in *Connecting to an External Database* on page 808), prefixing the encrypted password with `ENCRYPTED`.
For example, `databasePool.password=ENCRYPTED:rO0ABXQABkRFU2VkZXVy`.

Configuring the Lightweight JDBC Adapter

To configure the Lightweight JDBC adapter, you must specify field settings in the Application and in the GPM.

The Application Configuration

The following table describes the fields used to configure the Lightweight JDBC adapter in the Application:

Field	Description
Name (Config)	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Start a new business process (StartNewWorkFlow)	Whether to start a new business process. Valid values: <ul style="list-style-type: none">◆ This Lightweight JDBC adapter will start a new business process.◆ This Lightweight JDBC adapter will not start a new business process.
Run as User	Enter (or select from the list) the user ID to be associated with business process instances of this service.
Use 24 Hour Clock Display	Select to specify times for this schedule using the 24 hour clock. Leave blank to use 12 hour clock and AM and PM.

Field	Description
Schedule	<p>Information about scheduling the Lightweight JDBC adapter configuration to run and to start the specified business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, this service does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the month Valid values are the day of the month (including the last day of the month (LDM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. <p>Note: The Schedule field only displays as an option if you set the <i>Start a new business process</i> parameter to "This Lightweight JDBC adapter configuration starts a new business process".</p>
Business process (InitialWorkflowId)	Business process you want the Lightweight JDBC adapter to start, if any. Required if you set the <i>Start a new business process</i> parameter to "This Lightweight JDBC adapter configuration will start a new business process".
Pool Name	Select the database pool to be used.
XML Result Root Tag (result_name)	<p>The root tag element you want to appear in the XML document returned from the Lightweight JDBC adapter after a query.</p> <p>This document contains the results from the SQL query and becomes the primary document. For example, if you specify RootResultXmlElement as the name for the XMLResult Root Tag, the following results are displayed in the returned XML document for an ACTION query:</p> <pre><RootResultXmlElement></pre> <pre></RootResultXmlElement></pre>

Field	Description
XML Result Row Tag (row_name)	<p>Row tag XML element containing all the column tags returned from the SQL query. The column tag names are not configurable. The tag names are generated by the column name returned in the result set. Required.</p> <p>For example, if you specify ResultsFromUpdate as the name for the XML Result Row Tag, the following results are displayed in the returned XML document for an ACTION query.</p> <pre><ResultsFromUpdate></pre> <pre></ResultsFromUpdate></pre>
Query Type (query_type)	<p>Result set or the number of rows affected by an action query returned by the SQL statement. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ SELECT – Returns the results from a query. The results are bounded to the XML Result Root tag (result_name), and then each returned row is bounded to the XML Result Row Tag (row_name). Each column from the Select statement also forms an XML element with the element name matching the column name and the element data matching the data returned from the query. <p>For example, if you query for all customers in a database (SELECT * FROM Customer), the results returned might look like the following:</p> <pre><XMLResultRootTag></pre> <pre><XMLResultRowTag></pre> <pre><CUSTOMER_ID>1</CUSTOMER_ID></pre> <pre><CUSTOMER_NAME>Jane Doe</CUSTOMER_NAME></pre> <pre></XMLResultRowTag></pre> <pre><XMLResultRowTag></pre> <pre><CUSTOMER_ID>2</CUSTOMER_ID></pre> <pre><CUSTOMER_NAME>Joe User</CUSTOMER_NAME></pre> <pre></XMLResultRowTag></pre> <pre></XMLResultRootTag></pre> ◆ INSERT – Executes an insert query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above. ◆ UPDATE – Executes an update query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above. ◆ ACTION – Executes an INSERT, UPDATE, or DELETE query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above. ◆ Stored procedure/function – Applies to stored procedures and functions.

Field	Description
	<p>Notes for stored procedures and functions:</p> <ul style="list-style-type: none"> ◆ When defining a Lightweight JDBC adapter instance using the GPM, you must declare a Query type of Stored Procedure/Function. When defining a Lightweight JDBC adapter instance using the BPML, you must declare a Query type of PROCEDURE. ◆ You should specify the stored procedure on the SQL parameter in the Lightweight JDBC adapter instance definition using the JDBC syntax. For example: <p>where ?,?,? are the arguments passed in from the parameter specifications (param1-param20) defined in the Lightweight JDBC adapter instance definition.</p> ◆ You should specify the function on the SQL parameter in the Lightweight JDBC adapter instance definition using the JDBC syntax. For example: <p>where ? is the argument returned from the function. You must specify a parameter type that matches the format of the value returned from the function. For example:</p> ◆ There are limitations when using Oracle to call stored procedures. For information about these limitations, see <i>Calling Oracle Stored Procedures</i> on page 832.
SQL Statement (sql)	<p>Hard-coded SQL query that queries a database. It must be in valid SQL syntax.</p> <p>Note: You can specify the SQL query either using this field or when you add the Lightweight JDBC adapter instance in a business process.</p>

GPM Configuration

The following screen shows the graphical view of the GPM parameters for the Lightweight JDBC adapter. The inactive values were specified using the Lightweight JDBC adapter configuration. The active fields are

fields that cannot be configured in the Application or that are being overridden. There are no fields that must be configured on the Message From Service tab.

Name: LWJDBC - QueryForCustomerID
 Config: ExampleLWJDBCBusinessProcess

Message To Service | **Message From Service**

Output Msg: Obtain Message first, then Process Data
 Message Name: LightweightJDBCAdapterTypeInputMessage

Name	Value	Use XPath?
InitialWorkflowId		<input type="checkbox"/>
param1	number(ProcessData/CustomerInformation/ResultsFromQuery/Customer(number((NumberOfRecords))/customer_id/text()))	<input checked="" type="checkbox"/>
param10		<input type="checkbox"/>
param11		<input type="checkbox"/>
param12		<input type="checkbox"/>
param13		<input type="checkbox"/>
param14		<input type="checkbox"/>
param15		<input type="checkbox"/>
param16		<input type="checkbox"/>
param17		<input type="checkbox"/>
param18		<input type="checkbox"/>
param19		<input type="checkbox"/>
param2		<input type="checkbox"/>
param20		<input type="checkbox"/>
param3		<input type="checkbox"/>
param4		<input type="checkbox"/>
param5		<input type="checkbox"/>
param6		<input type="checkbox"/>
param7		<input type="checkbox"/>
param8		<input type="checkbox"/>

Name	Value	Use XPath?
param9		<input type="checkbox"/>
paramtype1	Integer	<input type="checkbox"/>
paramtype10		<input type="checkbox"/>
paramtype11		<input type="checkbox"/>
paramtype12		<input type="checkbox"/>
paramtype13		<input type="checkbox"/>
paramtype14		<input type="checkbox"/>
paramtype15		<input type="checkbox"/>
paramtype16		<input type="checkbox"/>
paramtype17		<input type="checkbox"/>
paramtype18		<input type="checkbox"/>
paramtype19		<input type="checkbox"/>
paramtype2		<input type="checkbox"/>
paramtype20		<input type="checkbox"/>
paramtype3		<input type="checkbox"/>
paramtype4		<input type="checkbox"/>
paramtype5		<input type="checkbox"/>
paramtype6		<input type="checkbox"/>
paramtype7		<input type="checkbox"/>
paramtype8		<input type="checkbox"/>

paramtype9		<input type="checkbox"/>
pool	mysqlTrainingPool	<input type="checkbox"/>
query_type	Select	<input type="checkbox"/>
result_name	RootResultXmlElement	<input type="checkbox"/>
row_name	ResultsFromSQL	<input type="checkbox"/>
sql	SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2...	<input type="checkbox"/>
StartNewWorkflow	This Lightweight JDBC Adapter will not start a new busine...	<input type="checkbox"/>

The following example shows the corresponding BPML parameters for the Lightweight JDBC adapter GPM parameters. This example business process takes the primary document, which contains the query results, and writes it to process data.

The following table describes the fields used to configure the Lightweight JDBC adapter in the GPM. This table contains only the fields that are configured in the GPM. See *The Application Configuration* on page 818 for parameters you can specify in the GPM or in the Application configuration.

Field	Description
Config	Name of the service configuration.
param1 - param20	Specifies the values that are passed to the SQL statement that is to be executed against the database. You can either specify constant values or use XPath query statements to retrieve information from process data to pass to the SQL statement. If you use XPath query statements, you must also select the "Use XPath?" check box in the Service Editor of the GPM for the parameter. See the <i>Lightweight JDBC Business Process Usage</i> on page 827 for an example of using Xpath to set these values. Optional.
paramtype1 - paramtype20	Parameter type that corresponds to the parameter number (for example, param1 and paramtype1). Every parameter specified must have a corresponding parameter type. Valid values: <ul style="list-style-type: none"> ◆ Cursor ◆ String ◆ CharacterStream ◆ CharacterStreamFromDocument ◆ Integer ◆ Long ◆ Double ◆ Float ◆ Date

Large Text Data Support

Two values are available for large text data (clob) support:

- ◆ **CharacterStream** is used to insert or update data to the database that is in process data.
- ◆ **CharacterStreamFromDocument** is used to insert or update data to the database from the primary document.

To insert data to the database or to update the database using the **CharacterStream** or **CharacterStreamFromDocument** values, there are two new query types: **INSERT** and **UPDATE**.

There is also a parameter that you can add to your business process manually (this parameter is not available through the GPM): `write_characterstream_to_document`. Valid values for this parameter are **YES** and **NO**. This parameter enables you to place the contents of large data inside of a document during a **SELECT** query, because XML has size limitations on `pcdata`. The primary document will still contain the result tree but the name of the document will replace the large data. This new parameter is optional and defaults to **NO** for backward compatibility.

Note: When inserting, updating, or selecting clobs from Oracle, ensure the `varDataClassName` in the `jdbc.properties.in` file is pointing to `OracleBlobVarData` for an external pool.

The Lightweight JDBC adapter will attempt to get a connection a set number of times. The number of retries is configurable by changing the `externalDBPoolRetries` property in the `noapp.properties.in` file.

The following example illustrates using the parameter to write the contents of the data returned from the database directly to a document:

`doc-1`

The following example illustrates a sample of what you would see in the primary document:

The parameter types are used by the Lightweight JDBC adapter to insert or update large text data into the database. The insert and update statements differ in how they are implemented, so you must specify the INSERT or UPDATE action type. Several examples follow.

Example 1

Updates the column with the contents of the primary document. In this example, the parameter type is `CharacterStreamFromDocument`, so the parameter value is not required; instead, the value is taken from the primary document.

Example 2

Updates a column with the string returned from an Xpath query against process data. In this example, the `CharacterStream` parameter type is used, so the parameter value is required.

Example 3

Inserts a row with the string returned from an Xpath query against process data.

Example 4

Writes the contents of the data returned from a database directly to a document (note that this is not the primary document). The primary document will still contain the XML result tree; however, the data in the document is replaced by a document name.

This is an example of what you should expect to see in the primary document:

This is an Oracle example for updating a clob column. The first parameter is used to find the row in the table. The second parameter will be used to update the value.

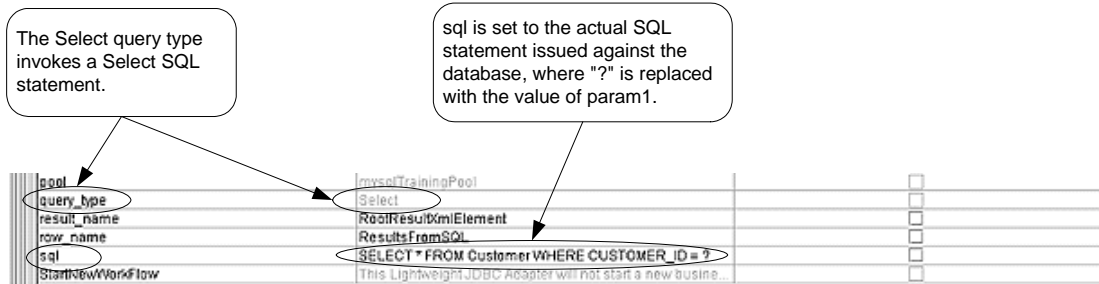
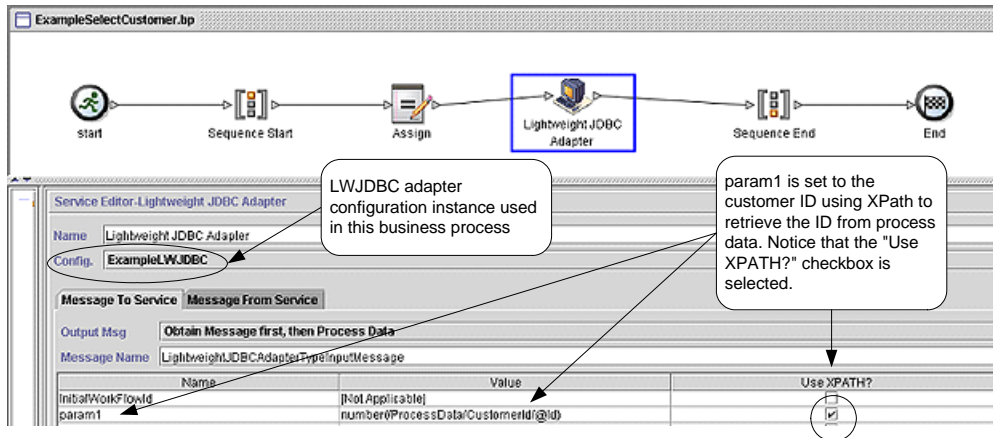
Lightweight JDBC Business Process Usage

This section contains additional examples using the Lightweight JDBC adapter, including sample configurations for using Oracle stored procedures and functions.

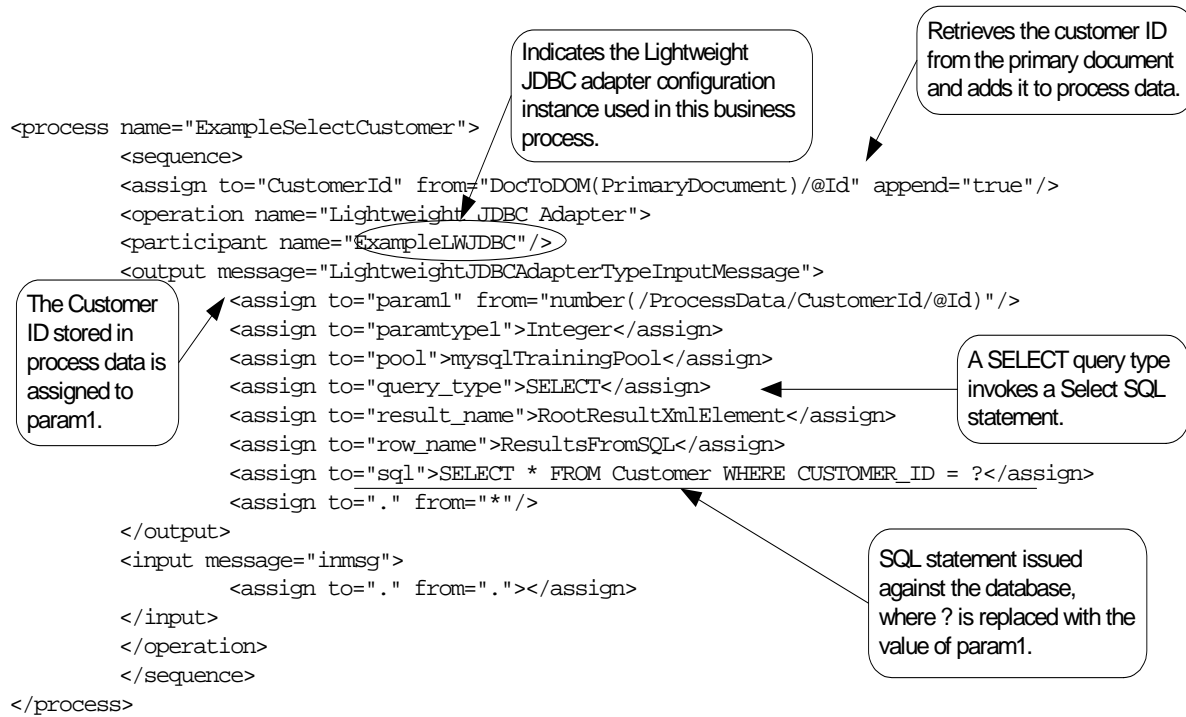
The following sample document is the primary document passed to the business process for the SELECT and ACTION query type examples.

SELECT Query Type Example

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to SELECT customer information based on the customer ID passed into the business process.



The following example illustrates the same business process using BPML.



Message to the Lightweight JDBC Adapter Instance

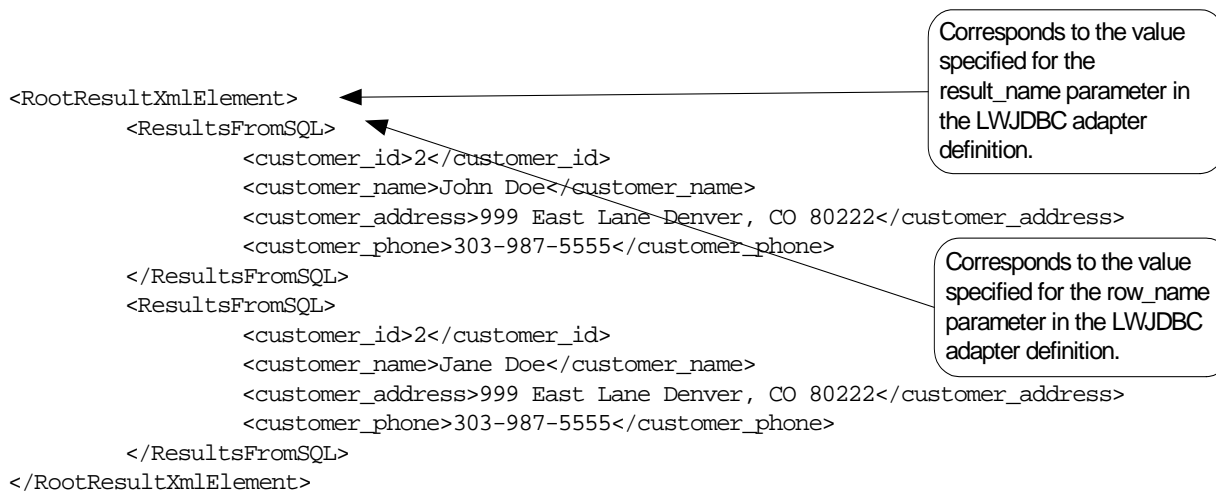
The following example shows the message sent to the Lightweight JDBC adapter when you run the business process. The message uses the parameters defined within the `<output>` tags of the Lightweight JDBC adapter definition in the previous BPML example, along with the parameters defined during configuration of the Lightweight JDBC adapter instance.

Note: You can view the message sent to the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message To Service”.

Message from the Lightweight JDBC Adapter Instance

The following example shows the table definition used to query against the Lightweight JDBC adapter. In the previous BPML example, the assign statement within the <input> tags of the Lightweight JDBC adapter definition (<assign to="." from="."></assign>) indicates that the results are returned as a primary document.

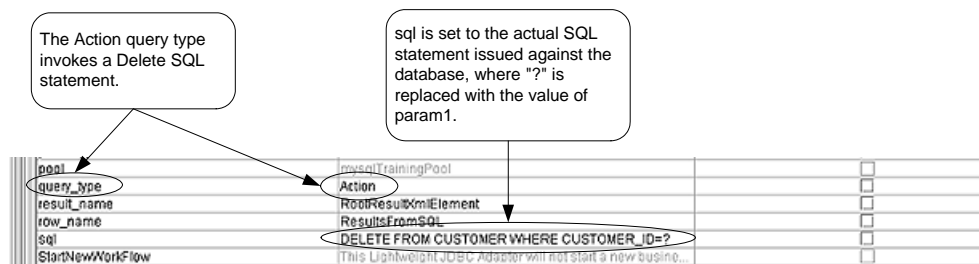
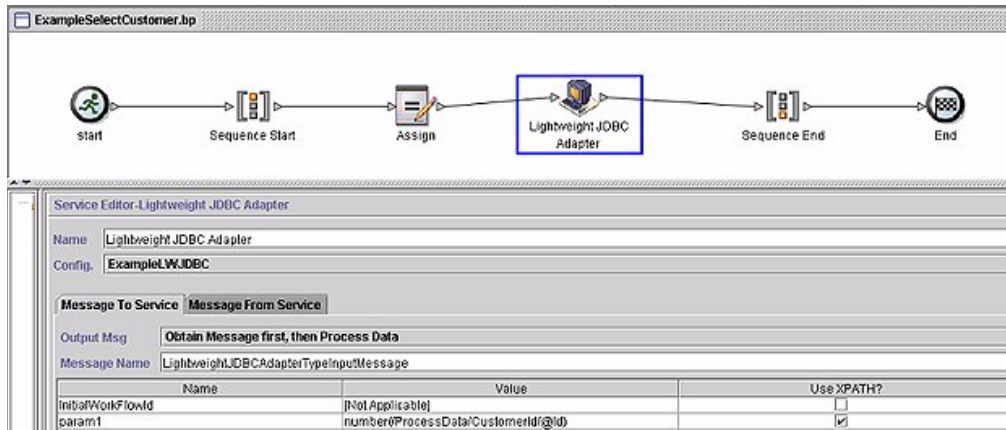
Notice how the columns from the Customer table become XML elements in the results returned from the Lightweight JDBC adapter, as shown below in the message returned from the Lightweight JDBC adapter.



Note: You can view the message returned from the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message From Service”.

ACTION Query Type Example

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to delete customer information from a database.



The following example illustrates the same business process using BPML.

```
<process name="ExampleDeleteCustomer">
  <sequence>
    <assign to="CustomerId" from="DocToDOM(PrimaryDocument)/@Id" append="true"/>
    <operation name="Lightweight JDBC Adapter">
      <participant name="ExampleLWJDBC"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="number(/ProcessData/CustomerId/@Id)"/>
        <assign to="paramtype1">Integer</assign>
        <assign to="query_type">ACTION</assign>
        <assign to="result_name">RootResultXmlElement</assign>
        <assign to="row_name">ResultsFromSQL</assign>
        <assign to="sql">DELETE FROM CUSTOMER WHERE CUSTOMER_ID=?</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="." />
      </input>
    </operation>
  </sequence>
</process>
```

The ACTION query type invokes a Delete SQL statement in this example.

Actual SQL statement issued against the database, where "?" is replaced with the value of param1.

Message to the Lightweight JDBC Adapter Instance

The following example shows the message sent to the Lightweight JDBC adapter when you run the business process. The message uses the parameters defined within the <output> tags of the Lightweight JDBC adapter definition in the previous BPML, along with the parameters defined for the Lightweight JDBC adapter instance during configuration.

Note: You can view the message sent to the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message To Service”.

Message from the Lightweight JDBC Adapter Instance

The following example shows the message returned from the Lightweight JDBC adapter to the business process. The assign statement within the <input> tags of the Lightweight JDBC adapter definition (<assign to="." from="."></assign>) indicates that the results are returned as a primary document.

```
<RootResultXmlElement>
<ResultsFromSQL>
<Rows_Affected>1</Rows_Affected>
</ResultsFromSQL>
</RootResultXmlElement>
```

The Lightweight JDBC adapter returns the number of rows deleted.

Note: You can view the message returned from the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message From Service”.

Calling Oracle Stored Procedures

This section describes limitations when using the Lightweight JDBC adapter to call Oracle stored procedures. These limitations apply only to Oracle databases.

The Lightweight JDBC adapter processes only one result set if you script multiple queries to run in stored procedures.

The first argument in an Oracle stored procedure must be declared as an output parameter, and the stored procedure must return a value to this parameter even if the value is not needed in the business process.

You must specify a value for the first parameter (param1) in the Lightweight JDBC adapter instance to correspond with the first argument of the stored procedure, even if the parameter is not used in the stored procedure.

Stored Procedure Example

The following example shows a database script for creating an Oracle stored procedure. The stored procedure updates the account status for a customer. Notice that the first argument of the stored procedure is declared as an output parameter and is set to a number, even though this value will not be used in the business process.

The following sample document is the primary document passed into the example business process used to call the stored procedure.

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to start the stored procedure.

Service Editor - Lightweight JDBC Adapter

Name: Lightweight JDBC Adapter
 Config: ExampleLWJDBCBusinessProcess

Message To Service / Message From Service

Output Msg: Obtain Message first, then Process Data
 Message Name: LightweightJDBCAdapterTypeInputMessage

Name	Value	Use XPath?
param1	1	<input type="checkbox"/>
param1 0		<input type="checkbox"/>
param1 1		<input type="checkbox"/>
param1 2		<input type="checkbox"/>
param1 3		<input type="checkbox"/>
param1 4		<input type="checkbox"/>
param1 5		<input type="checkbox"/>
param1 6		<input type="checkbox"/>

The value set for param1 is not used in the stored procedure.

param2	number(ProcessData/CustomerId/@id)	<input checked="" type="checkbox"/>
param3	number(ProcessData/Account/@id)	<input checked="" type="checkbox"/>
param4	ProcessData/AccountStatus/text()	<input checked="" type="checkbox"/>
param5		<input type="checkbox"/>
param6		<input type="checkbox"/>
param7		<input type="checkbox"/>
param8		<input type="checkbox"/>
param9		<input type="checkbox"/>
paramtype1	Integer	<input type="checkbox"/>
paramtype1 0		<input type="checkbox"/>
paramtype1 1		<input type="checkbox"/>
paramtype1 2		<input type="checkbox"/>
paramtype1 3		<input type="checkbox"/>
paramtype1 4		<input type="checkbox"/>
paramtype1 5		<input type="checkbox"/>

param2 is set to the customer Id.
 param3 is set to the customer account Id.
 param4 is set to the new account status.
 These values are used in the stored procedure to update the customer account status.

paramtype17		<input type="checkbox"/>
paramtype18		<input type="checkbox"/>
paramtype19		<input type="checkbox"/>
paramtype2	Integer	<input type="checkbox"/>
paramtype20		<input type="checkbox"/>
paramtype3	Integer	<input type="checkbox"/>
paramtype4	String	<input type="checkbox"/>
paramtype5		<input type="checkbox"/>
paramtype6		<input type="checkbox"/>
paramtype7		<input type="checkbox"/>
paramtype8		<input type="checkbox"/>
paramtype9		<input type="checkbox"/>
pool	OraclePool	<input type="checkbox"/>
query_type	PROCEDURE	<input type="checkbox"/>
result_name	RootResultXmlElement	<input type="checkbox"/>
row_name	ResultsFromSQL	<input type="checkbox"/>
sql	{call P_Update_Account(?,?,?)}	<input type="checkbox"/>
StartNewWorkflow	NO	<input type="checkbox"/>

The PROCEDURE query type invokes a stored procedure in this example.

sql is set to the stored procedure to invoke, where "?,?,?" is replaced with the values defined for param1, param2, param3, and param4.

The following example illustrates the same business process using BPML:

```

<process name="ExampleUpdateAccountStatus">
  <sequence>
    <assign to="CustomerId" from="DocToDOM(PrimaryDocument)/Account/Customer/@Id"
append="true"/>
    <assign to="Account" from="DocToDOM(PrimaryDocument)/Account/@Id" append="true"/>
    <assign to="AccountStatus" from="DocToDOM(PrimaryDocument)/Account/Status/text()"
append="true"/>
    <operation name="Lightweight JDBC Adapter">
      <participant name="ExampleLWJDBCBusinessProcess"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1">1</assign>
        <assign to="paramtype1">Integer</assign>
        <assign to="param2" from="number(/ProcessData/CustomerId/@Id)"/>
        <assign to="paramtype2">Integer</assign>
        <assign to="param3" from="number(/ProcessData/Account/@Id)"/>
        <assign to="paramtype3">Integer</assign>
        <assign to="param4" from="/ProcessData/AccountStatus/text()"/>
        <assign to="paramtype4">String</assign>
        <assign to="pool">oraclePool</assign>
        <assign to="query_type">PROCEDURE</assign>
        <assign to="result_name">RootResultXmlElement</assign>
        <assign to="row_name">ResultsFromSQL</assign>
        <assign to="sql">{call P_Update_Account(?,?,?,?)}</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="." />
      </input>
    </operation>
  </sequence>
</process>

```

Specify PROCEDURE as the query type when using BPML.

Notice the format used to call stored procedures.

Calling Oracle Functions

The following example shows a database script for creating a simple Oracle function. The function retrieves and returns the number of products in the database:

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to start the function:

ExampleGetProductCount.bp

Service Editor - Lightweight JDBC Adapter

Name: Lightweight JDBC Adapter
 Config: ExampleLVJDBCBusinessProcess

Message To Service: Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: LightweightJDBCAdapterTypeInputMessage

Name	Value	Use XPATH?
paramtype1	Integer	<input type="checkbox"/>
paramtype10		<input type="checkbox"/>
paramtype11		<input type="checkbox"/>
paramtype12		<input type="checkbox"/>
paramtype13		<input type="checkbox"/>
paramtype14		<input type="checkbox"/>
paramtype15		<input type="checkbox"/>
paramtype16		<input type="checkbox"/>
paramtype17		<input type="checkbox"/>
paramtype18		<input type="checkbox"/>
paramtype19		<input type="checkbox"/>

Advanced

No input parameters are specified for the function. However, you must specify a parameter type that matches the format of the value returned from the function. In this case, we set paramtype1 to Integer because the function returns the number of products.

paramtype19		<input type="checkbox"/>
paramtype2		<input type="checkbox"/>
paramtype20		<input type="checkbox"/>
paramtype3		<input type="checkbox"/>
paramtype4		<input type="checkbox"/>
paramtype5		<input type="checkbox"/>
paramtype6		<input type="checkbox"/>
paramtype7		<input type="checkbox"/>
paramtype8		<input type="checkbox"/>
paramtype9		<input type="checkbox"/>
pool	oraclePool	<input type="checkbox"/>
query_type	Stores procedure/function	<input type="checkbox"/>
result_name	TestIDBLookupResult	<input type="checkbox"/>
row_name	ResultsFromSQL	<input type="checkbox"/>
sql	{? = call product_count() }	<input type="checkbox"/>
StartNewWorkflow	This Lightweight JDBC Adapter will not start a new busine...	<input type="checkbox"/>

sql is set to the function to invoke, where "?" is replaced with the value returned from the function and mapped to paramtype1.

The following example illustrates the same business process using BPML:

```
<process name="ExampleGetProductCount">
  <sequence>
    <operation name="Lightweight JDBC Adapter">
      <participant name="ExampleLWJDBCBusinessProcess"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="paramtype1">Integer</assign>
        <assign to="pool">oraclePool</assign>
        <assign to="query_type">PROCEDURE</assign>
        <assign to="result_name">TestDBLookupResult</assign>
        <assign to="row_name">ResultsFromSQL</assign>
        <assign to="sql">{? = call product_count()}</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="." />
      </input>
    </operation>
  </sequence>
</process>
```

Specify PROCEDURE as the query type when using BPML.

Notice the format used to call functions.

Lightweight Java Database Connectivity (JDBC) Adapter (Build 4321 or higher)

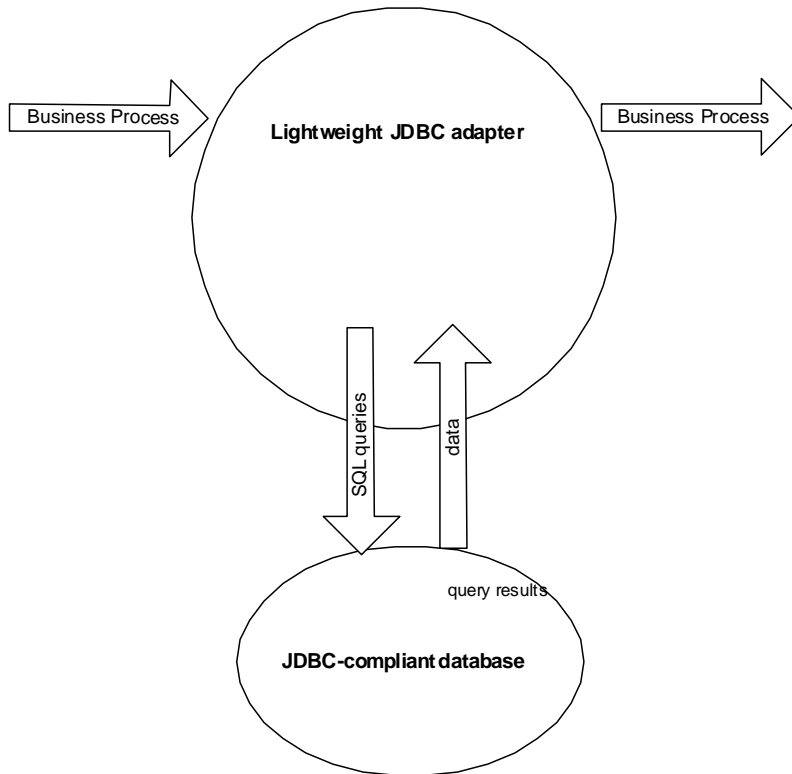
The following table provides a high-level overview of the Lightweight JDBC adapter:

System name	None
Graphical Process Modeler (GPM) category	All services, Translation
Description	<p>Enables you to retrieve data from a JDBC-compliant database or update a JDBC-compliant database as part of a business process within the Application, using one of the following methods:</p> <ul style="list-style-type: none">◆ Specify XPath queries to retrieve information from process data and set the Lightweight JDBC (LWJDBC) adapter parameters in a business process. Use XPath queries to set the Lightweight JDBC adapter parameters in a business process, allowing for reuse of the same instance of a Lightweight JDBC adapter configuration multiple times throughout a business process. To use XPath queries, you need to first load into process data the data which you want to use to set the Lightweight JDBC adapter parameters. You can use either the XML Encoder service or DocToDOM function in your business process to load into process data a primary document containing the data you want to use for XPath queries.◆ Specify constants in the Lightweight JDBC adapter configuration. Unlike the JDBC adapter, the Lightweight JDBC adapter does not use a map in which the SQL statements are fixed. The Lightweight JDBC adapter enables you to specify parameters in the SQL statement that are supplied at run time. For information about the JDBC adapter, see <i>Java Database Connectivity (JDBC) Adapter</i>. The Lightweight JDBC adapter executes a query, and you receive an XML document that contains the results. This document becomes the primary document in the business process. Then you can either:<ul style="list-style-type: none">◆ Load the XML document into process data.◆ Carry the XML document forward into another service in the business process.
Business usage	<p>Use the Lightweight JDBC adapter if you want to retrieve or manipulate data and need the flexibility of specifying various SQL queries.</p> <p>Use the JDBC adapter if you have complex output and you want to manipulate the data from a database and control the structure of the output.</p>
Preconfigured?	No
Requires third party files?	You must configure a connection to an external database for the Lightweight JDBC adapter.
Platform availability	All supported Application platforms
Related services	None
Initiates business processes?	Yes
Invocation	Runs by a schedule or business process.

How the Lightweight JDBC Adapter Works

The Lightweight JDBC adapter can start a business process or it can be used in a business process. This service can be scheduled to run at weekly or timed intervals.

The following figure shows how the Lightweight JDBC adapter communicates with a JDBC-compliant database:



Business Scenario

Your company receives information from an external database about a customer that needs to be either added (if the customer does not exist) or updated in your internal customer database. The customer information arrives in XML format.

Business Solution Example

The following approach is used to solve the business scenario:

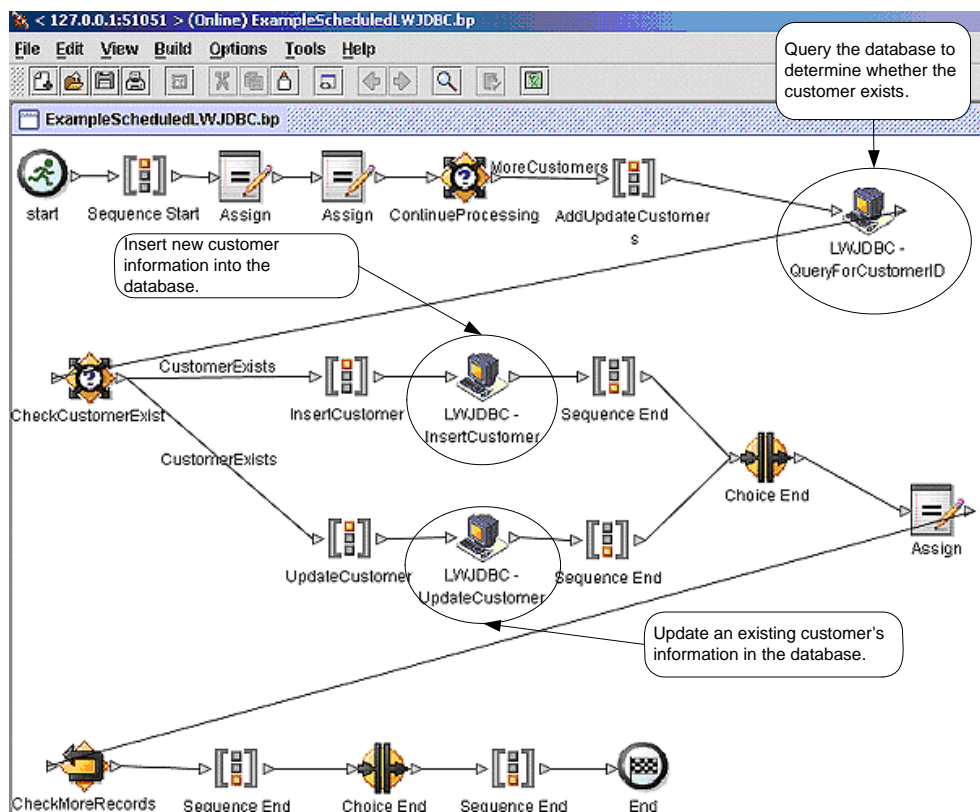
1. Configure a Lightweight JDBC adapter instance to be included in a business process. This one instance of the Lightweight JDBC adapter is used multiple times throughout the example business process to perform the following tasks:
 - ◆ Query the database for the customer ID passed into the business process to determine if the customer currently exists in the internal customer database.
 - ◆ Add the customer information to the customer database if the customer does not exist.
 - ◆ Update the customer information in the customer database if the customer does exist.

2. Create a business process that determines whether to add or update the customer information to the internal customer database. The Lightweight JDBC adapter instance configured above is used in this business process to perform the query, add, and/or update.
3. Configure a second Lightweight JDBC adapter instance to retrieve customer records from the external database and to start the business process created in the previous step. This Lightweight JDBC adapter instance is also scheduled to run at 10 p.m. daily to retrieve customer records from the external database and return the results to the business process.

GPM Example

The following example shows a solution to the business scenario using the GPM.

A Lightweight JDBC adapter has been configured and scheduled to retrieve records from an external database. The result from the query is the primary document to this business process.



The Assign statements are used to retrieve the customer information from the primary document and place the information into process data. The parameters specified for the Lightweight JDBC adapter, which includes customer information retrieved from process data and SQL statements, are input to the adapters. A primary document containing the results from the SQL statements issued against the database is output from the Lightweight JDBC adapter instances and is passed to the next step in the business process.

For information about the Lightweight JDBC parameters, see *Implementing the Lightweight JDBC Adapter* on page 842.

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML.

```
<process name="ExampleScheduledLWJDBC">
<rule name="MoreCustomers">
  <condition>NumberOfRecords > 0</condition>
</rule>
<rule name="CustomerExists">
  <condition>number(/ProcessData/NumberOfCustomers) = 0 </condition>
</rule>
<sequence>
  <assign to="NumberOfRecords" from="number(count(DocToDOM(PrimaryDocument)/Customer/
customer_id))" append="true"/>
  <assign to="CustomerInformation" from="DocToDOM(PrimaryDocument)" append="true"/>
  <choice name="ContinueProcessing">
    <select>
      <case ref="MoreCustomers" activity="AddUpdateCustomers"/>
    </select>
    <sequence name="AddUpdateCustomers">
      <operation name="LWJDBC - QueryForCustomerID">
        <participant name="ExampleLWJDBCBusinessProcess"/>
        <output message="LightweightJDBCAdapterTypeInputMessage">
          <assign to="param1" from="number(/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_id/text())"/>
          <assign to="paramtype1">Integer</assign>
          <assign to="query_type">SELECT</assign>
          <assign to="result_name">RootResultXmlElement</assign>
          <assign to="row_name">ResultsFromSQL</assign>
          <assign to="sql">SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2 WHERE CUSTOMER_ID = ?</
assign>
          <assign to="." from="*" />
        </output>
        <input message="inmsg">
          <assign to="NumberOfCustomers" from="DocToDOM(PrimaryDocument)/ResultsFromSQL/
CUSTOMERS" append="true"/>
        </input>
      </operation>
      <choice name="CheckCustomerExist">
        <select>
          <case ref="CustomerExists" activity="InsertCustomer"/>
          <case ref="CustomerExists" negative="true" activity="UpdateCustomer"/>
        </select>
        <sequence name="InsertCustomer">
          <operation name="LWJDBC - InsertCustomer">
            <participant name="ExampleLWJDBCBusinessProcess"/>
            <output message="LightweightJDBCAdapterTypeInputMessage">
              <assign to="param1" from="number(/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_id/text())"/>
              <assign to="param2" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_name/text()"/>
              <assign to="param3" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_address/text()"/>
              <assign to="param4" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(/NumberOfRecords)]/customer_phone/text()"/>
            </output message>
          </operation>
        </sequence>
      </choice>
    </sequence>
  </choice>
</sequence>
```

The LWJDBC instance used to determine if the customer exists

The LWJDBC instance used to add a customer to the database

```

    <assign to="paramtype1">Integer</assign>
    <assign to="paramtype2">String</assign>
    <assign to="paramtype3">String</assign>
    <assign to="paramtype4">String</assign>
    <assign to="query_type">ACTION</assign>
    <assign to="result_name">RootResultXmlElement</assign>
    <assign to="row_name">ResultsFromInsert</assign>
    <assign to="sql">INSERT INTO CUSTOMER2 VALUES(?, ?, ?, ?)</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="." />
  </input>
</operation>
</sequence>
<sequence name="UpdateCustomer">
  <operation name="LWJDBC - UpdateCustomer">
    <participant name="ExampleLWJDBCBusinessProcess" />
    <output message="LightweightJDBCAdapterTypeInputMessage">
      <assign to="param1" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_name/text()" />
      <assign to="param2" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_address/text()" />
      <assign to="param3" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_phone/text()" />
      <assign to="param4" from="number(/ProcessData/CustomerInformation/ResultsFromQuery.
Customer[number(//NumberOfRecords)]/customer_id/text())" />
      <assign to="paramtype1">String</assign>
      <assign to="paramtype2">String</assign>
      <assign to="paramtype3">String</assign>
      <assign to="paramtype4">Integer</assign>
      <assign to="query_type">ACTION</assign>
      <assign to="result_name">RootResultXmlElement</assign>
      <assign to="row_name">ResultsFromUpdate</assign>
      <assign to="sql">UPDATE CUSTOMER2 SET CUSTOMER_NAME = ?, CUSTOMER_ADDRESS=?,
CUSTOMER_PHONE=? where CUSTOMER_ID=?</assign>
      <assign to="." from="*" />
    </output>
    <input message="inmsg">
      <assign to="." from="." />
    </input>
  </operation>
</sequence>
</choice>
<assign to="NumberOfRecords" from="NumberOfRecords - 1" />
<repeat name="CheckMoreRecords" ref="ContinueProcessing" />
</sequence>
</choice>
</sequence>
</process>

```

The LWJDBC instance used to update an existing customer's information in the database

For information about the Lightweight JDBC parameters, see *Implementing the Lightweight JDBC Adapter* on page 842. For additional Lightweight JDBC adapter examples, including examples for using Oracle stored procedures and functions, see *Lightweight JDBC Business Process Usage* on page 864.

Implementing the Lightweight JDBC Adapter

You can implement the Lightweight JDBC adapter by:

Configuring a Lightweight JDBC adapter instance to include in business processes. This lets you reuse the same Lightweight JDBC adapter instance to run different SQL queries against a database.

Configuring a Lightweight JDBC instance that runs an SQL query and then starts a business process.

The information in this section applies to both implementation types.

To implement the Lightweight JDBC adapter, complete the following tasks:

1. Collect the following information:
 - ◆ The name of the business process (if the adapter is to start a business process)
 - ◆ Database pool name used to connect to an external database
 - ◆ Parameter values for the `jdbc_customer.properties.in` file (see *Adding New Database Pools* on page 843).
2. Set up a connection to an external database, if you have not already done so. For information, see *Setting Up a Connection to an External Database* on page 843.
3. Create a Lightweight JDBC adapter configuration. For information, see *Managing Services and Adapters*.
4. Configure the Lightweight JDBC adapter. For information, see *Configuring the Lightweight JDBC Adapter* on page 854.
5. Use the Lightweight JDBC adapter in a business process.

Note: If the Lightweight JDBC adapter configuration starts a business process, create the business process prior to configuring the adapter.

Setting Up a Connection to an External Database

You must set up a connection to an external database for the Lightweight JDBC adapter. You can use any of the databases supported by the Application for internal use or other JDBC-compliant databases, such as Sybase. For information about databases supported by the Application, see *System Requirements*.

Adding New Database Pools

To define a new database pool for use by the Lightweight JDBC adapter, you must add settings for the pool to the `jdbc_customer.properties.in` file, which is located in the Application `/install_dir/properties` directory.

In `jdbc_customer.properties.in`, specify the database server name, port number, database/catalog name, user ID and password. To encrypt your database password, use the `encrypt_string.sh` or `encrypt_string.cmd` utility in the `bin` directory. Then place the encrypted password, prefixed by an encryption indicator, in your properties file.

Caution: There are two `jdbc_customer.properties` files: `jdbc_customer.properties.in`, which is the “template” properties file; and `jdbc_customer.properties`, which is the “packaged” properties file.

It is extremely important to ensure that you add the records to the template file, `jdbc_customer.properties.in`, not to the packaged file.

Each time you run the `setupfiles` command in the Application, all packaged files are updated with

the information contained in their template (.in) files. This means that if you make changes to the packaged file, `jdbc_customer.properties`, they are lost each time `setupfiles` is run. Always make changes to the template file, `jdbc_customer.properties.in`, and your changes will be maintained.

If the database you want to connect to resides on a database server type that is not the same as the Application database server type, you also need to install a JDBC driver using the `install3rdparty.sh` or `install3rdparty.cmd` utility.

Select a table and column in your database to use in the test on reserve function. This function causes the Application to test the database connection using a quickly run query before attempting to use it. This function ensures that idle connections are revived. The column referenced in the query should be of the type *varchar* and should be at least five characters in length.

You can change database passwords on a scheduled basis in Sterling Integrator. When you add or modify a pool, you now can control user and password credentials with effective dates. Multiple user and password credentials are associated with a pool. A date/time entry indicates to Sterling Integrator when to start using that credential for new connections. This applies primarily to external pools, although Sterling Integrator database pools will also work.

You can use the following variables for the date format:

```
15:00:00 3/16/09
3/16/09 15:00:00
3/16/2009 15:00:00
Sat, 12 Aug 1995 13:30:00 GMT
Sat, 12 Aug 1995 13:30:00 GMT+0430
```

Other formats may be used as long as they follow the Internet Engineering Task Force (IETF) standard date syntax. For additional information see <http://www.ietf.org/rfc/rfc3339.txt>

Pool Property	Description
<code>newDBPool.password.1=</code> <code><new password></code>	You can specify alphabets and combination of alphabets and numbers for the password. You can use numbers for <code>newDBPool.password.1</code> or <code>newDBPool.password.2</code> as well as following examples: <code>newDBPool.password.a=password_a</code> <code>newDBPool.effective.a=1/01/2005 09:35:00</code> <code>newDBPool.password.b=password_b</code> <code>newDBPool.effective.b=02/01/2009 09:35:00</code> <code>newDBPool.password.c=password_c</code> <code>newDBPool.effective.c=06/18/2009 11:07:00</code>

Pool Property	Description
newDBPool.effective.1= <The date for the new password starts to take affect>	<p>You can specify alphabets and combination of alphabets and numbers for the password. You can use numbers for <i>newDBPool.password.1</i> or <i>newDBPool.password.2</i> as well as following examples:</p> <pre>newDBPool.password.a=password_a newDBPool.effective.a=1/01/2005 09:35:00 newDBPool.password.b=password_b newDBPool.effective.b=02/01/2009 09:35:00 newDBPool.password.c=password_c newDBPool.effective.c=06/18/2009 11:07:00</pre>

Connecting to an External Database

To connect to an external database:

1. Add the necessary records to the `jdbc_customer.properties.in` file found in the `/install_dir/properties` directory.

Note: If invalid data (like ABC or 13 . 45) is entered in a pool setting, the setting uses its default value.

See the examples that follow this procedure for *Oracle 8i/9i* on page 852, *DB2* on page 852, *MS SQL 2000* on page 853, and *Sybase* on page 854.

The following table contains the parameters needed to add a new database pool to the `jdbc_customer.properties.in` file:

Parameter	Description
<code>databasePool.driver</code>	JDBC driver class file for the database application.
<code>databasePool.url</code>	<p>Database location (full URL as defined by the Java JDBC standards).</p> <p>Note: For Oracle systems, the last segment in the URL is the Oracle SID (not the System Reference or Tnsnames entry).</p> <p>Note: You can locate the Java JDBC standards on the java.sun.com Web site.</p>
<code>databasePool.user</code>	Username for logging into the database.
<code>databasePool.password</code>	Password for logging into the database.
<code>databasePool.maxconn</code>	Maximum number of database connections for the connection pool.

Parameter	Description
<i>databasePool.storedProcClassName</i>	<p>Specifies the class that handles stored procedure calls for the Lightweight JDBC adapter.</p> <p>The following classes are used for the database types:</p> <ul style="list-style-type: none"> ◆ MSSQL, Sybase, and DB2 – com.sterlingcommerce.woodstock.util.frame.jdbc.GenericStoredProcQuery ◆ Oracle 8i/9i – com.sterlingcommerce.woodstock.util.frame.jdbc.OracleNoAppStoredProcQuery <p>Note: The Lightweight JDBC adapter does not support stored procedures for MySQL.</p>
<i>databasePool.varDataClassName</i>	<p>Each database that the Application supports handles binary objects differently. This parameter specifies the class used to handle binary data for the database. Enter the correct class for your database:</p> <ul style="list-style-type: none"> ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ISeriesVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2VarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ZOSVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.JConnectVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.MSSQLVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.MySQLVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.OracleBlobVarData ◆ com.sterlingcommerce.woodstock.util.frame.jdbc.OracleVarData
<i>databasePool.catalog</i>	Database name (usually the same as the last segment of the URL)
<i>databasePool.type</i>	<p>Valid values:</p> <ul style="list-style-type: none"> ◆ local ◆ remote (default)
<i>databasePool.testOnReserve</i>	<p>Whether to test the connection.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ true ◆ false <p>Note: This function causes the Application to test the database connection before attempting to use it and revives idle connections.</p>

Parameter	Description
<i>databasePool.testOnReserveQuery</i>	<p>SQL query to use when testing the connection. Select a table and column in your database to use in the test on reserve function. The column referenced in the query should be of the type <i>varchar</i> and should be at least five characters in length. This query needs to be executable by the <i>databasePool.username</i> account and must be a valid SQL query. For example:</p> <pre>SELECT table_name FROM user_tables WHERE table_name=?</pre> <p>where ? must accept a string value. The query does not have to return a value to operate. If the query fails, the Database Pool is not activated.</p>
<i>databasePool.testOnReserveInterval</i>	<p>The minimum number of milliseconds between running testOnReserve on the same connection. The default value is 60000. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 - No interval and current interval is used. ◆ <= 0 - No interval. ◆ > 0 - The minimum number of milliseconds between running testOnReserve on the same connection.
<i>databasePool.max8177RetryCount</i>	<p>Only used for an Oracle database, this tells the software how many times to retry if it receives an ORA-8177 error in certain situations.</p>
<i>databasePool.dbvendor</i>	<p>Enter the database name: sybase, oracle, mysql, mssql, db2, db2zos, db2iseries, or other vendor name.</p>
<i>databasePool.maxsize</i>	<p>Maximum size of the database pool. This property was previously contained in the poolManager.properties file. This value must not exceed the value specified for the databasePool.maxconn parameter in the jdbc.properties file.</p>
<i>databasePool.initsize</i>	<p>Initial size of the database pool. This property was previously contained in the poolManager.properties file.</p>
<i>databasePool.factory</i>	<p>Always enter the following: com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory</p>

Parameter	Description
<i>databasePool.behaviour</i>	<p>Behavior a connection pool exhibits when it runs out of connections. This property replaces the <i>databasePool.onEmpty</i> property in the former <i>poolManager.properties</i> file. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 – The pool simply returns indicating to the software to abort its current action and try again later. This value corresponds to the value return in the <i>databasePool.onEmpty</i> property. ◆ 1 – The pool waits the number of milliseconds specified in <i>databasePool.waittime</i> for a connection to be returned before indicating to the software to abort and try again. This value corresponds to the value <i>wait</i> in the <i>databasePool.onEmpty</i> property. ◆ 2 – The pool creates a buffered connection (a connection above the size specified in <i>databasePool.maxsize</i>). When using a setting of 2, the maximum number of connections for the pool is the value specified for <i>databasePool.maxsize</i> plus the value specified for <i>databasePool.buffersize</i> connections. This allows connections to be created under heavy demand. This value corresponds to the value <i>new</i> in the <i>databasePool.onEmpty</i> property.
<i>databasePool.lifespan</i>	<p>The number of milliseconds a connection will live in a given pool before it needs to be removed.</p> <ul style="list-style-type: none"> ◆ 0 - (Default) No timeout. ◆ <= 0 - No timeout. ◆ > 0 - Number of milliseconds that a connection stay in pool.
<i>databasePool.idletimeout</i>	<p>The number of milliseconds a connection can stay idle in a given pool before it needs to be removed. The default value is 86400000. Valid values:</p> <ul style="list-style-type: none"> ◆ 0 - No timeout. ◆ <= 0 - No timeout. ◆ > 0 - Number of milliseconds that a connection stay in pool.
<i>databasePool.housekeepinginterval</i>	<p>The minimum number of milliseconds between running the housekeeping task to clean out idle connections. Valid values are any positive number. The default value is 3600000 milliseconds (1 hour). Any number less than 3600000 will cause the default of 3600000 milliseconds to be used.</p>
<i>databasePool.buffersize</i>	<p>Number of extra connections that the connection pool can create above the value specified for <i>databasePool.maxsize</i> to improve handling of unanticipated loads on the system. This property is only used if <i>databasePool.behavior</i> is set to 2.</p>
<i>databasePool.waittime</i>	<p>Amount of time (in milliseconds) to wait for a connection to become available before indicating to the software to abort the current action and try again later. This property is only used if <i>databasePool.behavior</i> is set to 1.</p>

2. Run the *setupfiles.sh* (UNIX) or *setupfiles.cmd* (Windows) utility located in the */install_dir/bin* directory of the Application installation directory. This updates the “packaged” properties file, *jdbc_customer.properties*, with the changes from the “template” properties file, *jdbc_customer.properties.in*.

3. If the vendor for the connection database is not the same vendor as the database vendor used for the Application database, install the appropriate JDBC driver to access the database server. Use the `install3rdParty.sh` (UNIX) or `install3rdparty.cmd` (Windows) utility located in the `/install_dir/bin` directory of the Application installation directory to add the JDBC driver jar file(s). Type **install3rdParty** on the command line to get a description of the parameters you can specify.

The following examples are for a UNIX environment. The vendor name and version are the first two parameters, along with the location of the zip file containing the JDBC driver files.

- ◆ For Oracle 9i, install the driver using the following command:

```
./install3rdParty.sh Oracle 9_2_0_5 -d  
/usr local directory/oracle/9_2_0_5/classes12.zip
```

- ◆ For DB2, install the driver using the following command:

```
./install3rdParty.sh db2java 7_2 -d / / / /db2java.zip
```

- ◆ To install the jConnect driver for Sybase, refer to *Installing a Sybase Driver* on page 849.

4. Stop and restart the Application to use the changed files.

Installing a Sybase Driver

Install the jConnect driver for Sybase using the following procedure:

1. Download `jConnect-5_5.zip` from the Sybase web site.
2. Run the following command:

```
./install3rdParty.sh jconnect 5_5 -d  
/ / / /5_5/jConnect-5_5.zip
```

- ◆ If this command succeeds, you are finished with this procedure.
- ◆ If Application reports in the system log that the driver could not be registered because the driver class cannot be found, continue with the procedure. Use the following steps to remove existing references to jConnect.

3. Stop Application.
4. Change your directory to `install_dir/jar`.
5. Delete any existing folders referencing jConnect.
6. Change your directory to `install_dir/properties`.
7. Open the files `dynamicclasspath.cfg` and `dynamicclasspath.cfg.in`. Delete any lines referencing jConnect, and save the files.
8. Create the following temporary directory:

```
install_dir/bin/jconnect
```

9. Extract only the jar files from `jConnect-5_5.zip` to this directory.

10. Run the following command:

```
install3rdparty jconnect 5_5 -d install_dir/bin/jconnect/*.jar
```

11. Check `install_dir/jar/jconnect/5_5/your_platform` to make sure that six jar files have been copied successfully.

12. Open the `dynamicclasspath.cfg` file in `install_dir/properties` and check that the following entries are there:

```
VENDOR_JAR=/          /jar/jconnect/5_5/your_platform
            install_dir          your_platform
            install_dir          your_platform/jisql.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2d.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/ribo.jar
```

13. Open the `customer.jdbc.properties.in` file in `install_dir/properties` and check that the following entries are there:

```
jconnectPool.driver=com.sybase.jdbc2.jdbc.SybDriver
jconnectPool.url=jdbc:sybase:Tds:your Hostname:4100/your DB
jconnectPool.user=your user name
jconnectPool.password=your password
jconnectPool.catalog=your database

jconnectPool.testOnReserveQuery=your Test On Reserve Query

jconnectPool.bufferSize=500
jconnectPool.maxSize=28
jconnectPool.initSize=1
jconnectPool.behaviour=2
jconnectPool.waitTime=1000
jconnectPool.storedProcClassName=
com.sterlingcommerce.woodstock.util.frame.jdbc.SybaseStoredProcQuery
jconnectPool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc
c.JConnectVarData
jconnectPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactory
```

14. Save the `customer.jdbc.properties.in` file and run the following command:

```
/bin/setupfiles.sh
```

This procedure should result in a successful connection to your Sybase database. However, if the database has been configured as character set `ROMAN8`, it is likely that you will see the following message in the Application system log, because of a limitation in the Sybase driver:

```
java.sql.SQLException: JZ0IB: The server's default charset of roman8 does not map
to an encoding that is available in the client Java environment. Because jConnect
will not be able to do client-side conversion, the connection is unusable and
is being closed. Try using a later Java version or try including your Java
installation's i18n.jar or charsets.jar file in the classpath
```

One resolution of this issue is to configure the primary Adaptive Server with a default character set that maps to one of the character sets supported by jConnect for JDBC (for example, UTF-8). For more details, refer to the Sybase documentation.

Another resolution of this issue is to use the open source jTDS driver from Sourceforge (sourceforge.net). To install this driver, follow these instructions:

1. Stop Application.
2. Remove references to jConnect as described previously.
3. Copy the jtds-1.2.jar file to an accessible directory on the Application machine.
4. Run the following command:

```
/bin/Install13rdparty.sh jTDS 1_2 - jar absolutePath
```

5. Check that the dynamicclasspath.cfg file has picked up this change. For example, */install_dir/jar/jTDS/1_2/your_platform/jtds-1.2.jar*.
6. Edit the jdbc_customer.properties.in file. The definition of the pool should be similar to the following example:

```
your user name  
your password  
your database
```

```
#jTDSPool.testOnReserveQuery=  
jTDSPool.dbvendor=jtds  
jTDSPool.bufferSize=50  
jTDSPool.maxSize=20  
jTDSPool.iniSize=5  
jTDSPool.behaviour=2  
jTDSPool.storedProcClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.  
.SybaseStoredProcQuery  
jTDSPool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.Ge  
nericVarData  
jTDSPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionF  
actory
```

7. Restart Application.

Properties File Examples for Specific Databases

Oracle 8i/9i

For Oracle 8i/9i, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=oracle.jdbc.driver.OracleDriver
           .url=jdbc:oracle:thin:@servername:0000:servername
databasePool.user=username
databasePool.password=password
databasePool.catalog=catalogname
databasePool.type=local
           .testOnReserve=
           .testOnReserveQuery=SELECT TestConnection from Connection_tb WHERE
TestConnection = ?
databasePool.testOnReserveInterval=60000
           .max8177RetryCount=
           .dbvendor=oracle
           .bufferSize=
           .maxSize=
           .initSize=
           .behaviour=
           .lifespan=0
           .idleTimeout=86400000
           .housekeepingInterval=3600000
           .waitTime=
           .storedProcClassName=
com.sterlingcommerce.woodstock.util.frame.jdbc.OracleNoAppStoredProcQuery
           .varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.OracleV
arData
           .factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactor
y
```

DB2

For DB2, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

Note: The JDBC adapter does not support stored procedures for DB2/iSeries and DB2/zOS.

```
Pool.driver=com.ibm.as400.access.AS400JDBCdriver
Pool.url=jdbc:as400://serverName/DB2Database;translate
binary=true;transaction isolation=none;
Pool.catalog=DB2Database
Pool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.DB2ISer
iesVarData
Pool.user=username
Pool.password=password
Pool.maxconn=20
Pool.testOnReserve=true
```

```

Pool.testOnReserveQuery=SELECT SI_VERSION from SI_VERSION where SI_VERSION =
?
Pool.testOnReserveInterval=60000
Pool.blobPageSize=1024000
Pool.compressBlob=true
Pool.cachePds=true
Pool.dbVendor=db2iSeries
Pool.bufferSize=500
Pool.maxSize=20
Pool.initSize=0
Pool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactor
Y
Pool.behaviour=2
Pool.lifespan=0
Pool.idleTimeout=86400000
Pool.housekeepingInterval=3600000
Pool.waitTime=1000

```

MS SQL 2000

For MS SQL 2000, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```

.driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
.url=jdbc:microsoft:sqlserver://servername:0000;DatabaseName=SQLdatabase
;SelectMethod=cursor
databasePool      username
databasePool      password
databasePool      catalogname
databasePool
databasePool      true
databasePool      SELECT TestConnection from Connection_tb WHERE
TestConnection = ?
databasePool
databasePool
databasePool      n
databasePool      n
databasePool      n
databasePool      n
databasePool
databasePool
databasePool      n
databasePool
databasePool
.com.sterlingcommerce.woodstock.util.frame.jdbc.GenericStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.jdbc.MSSQLVa
rData
.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.ConnectionFactor
Y

```

Sybase

For Sybase, enter the following parameters in the `jdbc_customer.properties.in` file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
.driver=com.sybase.jdbc2.jdbc.SybDriver
.url=jdbc:sybase:Tds:servername:0000/SybaseDB
databasePool      username
databasePool      password
databasePool      catalogname
databasePool
databasePool      true
databasePool      SELECT TestConnection from Connection_tb WHERE
TestConnection = ?
databasePool
databasePool
databasePool      n
databasePool      n
databasePool      n
databasePool      n
databasePool
databasePool
databasePool
databasePool      n
databasePool
databasePool
```

databasePool

databasePool

Encrypting Your Database Password

To use encryption for the database password:

1. Use `encrypt_string.sh` (UNIX) or `encrypt_string.cmd` (Windows).
2. When prompted, enter your external database password.
The script returns the encrypted value for your password.
3. Place the encrypted password in your `jdbc_customer.properties.in` file entry (see *Setting Up a Connection to an External Database* on page 843), prefixing the encrypted password with `ENCRYPTED`.

For example, `databasePool.password=ENCRYPTED:r00ABXQABkRFU2VkZXVy`.

Configuring the Lightweight JDBC Adapter

To configure the Lightweight JDBC adapter, you must specify field settings in the Application and in the GPM.

The Application Configuration

The following table describes the fields used to configure the Lightweight JDBC adapter in the Application:

Field	Description
Name (Config)	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Start a new business process (StartNewWorkFlow)	Whether to start a new business process. Valid values: <ul style="list-style-type: none">◆ This Lightweight JDBC adapter will start a new business process.◆ This Lightweight JDBC adapter will not start a new business process.
Run as User	Enter (or select from the list) the user ID to be associated with business process instances of this service.
Use 24 Hour Clock Display	Select to specify times for this schedule using the 24 hour clock. Leave blank to use 12 hour clock and AM and PM.

Field	Description
Schedule	<p>Information about scheduling the Lightweight JDBC adapter configuration to run and to start the specified business process.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, this service does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the month Valid values are the day of the month (including the last day of the month (LDM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. <p>Note: The Schedule field only displays as an option if you set the <i>Start a new business process</i> parameter to "This Lightweight JDBC adapter configuration starts a new business process".</p>
Business process (InitialWorkflowId)	Business process you want the Lightweight JDBC adapter to start, if any. Required if you set the <i>Start a new business process</i> parameter to "This Lightweight JDBC adapter configuration will start a new business process".
Pool Name	Select the database pool to be used.
XML Result Root Tag (result_name)	<p>The root tag element you want to appear in the XML document returned from the Lightweight JDBC adapter after a query.</p> <p>This document contains the results from the SQL query and becomes the primary document. For example, if you specify <code>RootResultXmlElement</code> as the name for the XML Result Root Tag, the following results are displayed in the returned XML document for an ACTION query:</p> <pre> <RootResultXmlElement> <ResultsFromUpdate> <Rows_Affected> 1</Rows_Affected> </ResultsFromUpdate> </RootResultXmlElement> </pre>

Field	Description
XML Result Row Tag (row_name)	<p>Row tag XML element containing all the column tags returned from the SQL query. The column tag names are not configurable. The tag names are generated by the column name returned in the result set. Required.</p> <p>For example, if you specify ResultsFromUpdate as the name for the XML Result Row Tag, the following results are displayed in the returned XML document for an ACTION query.</p> <pre><RootResultXmlElement> <ResultsFromUpdate> </ResultsFromUpdate></pre>
Query Type (query_type)	<p>Result set or the number of rows affected by an action query returned by the SQL statement. Required.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ SELECT – Returns the results from a query. The results are bounded to the XML Result Root tag (result_name), and then each returned row is bounded to the XML Result Row Tag (row_name). Each column from the Select statement also forms an XML element with the element name matching the column name and the element data matching the data returned from the query. <p>For example, if you query for all customers in a database (SELECT * FROM Customer), the results returned might look like the following:</p> <pre><XMLResultRootTag> <XMLResultRowTag> <CUSTOMER_ID>1</CUSTOMER_ID> <CUSTOMER_NAME>Jane Doe</CUSTOMER_NAME> </XMLResultRowTag> <XMLResultRowTag> <CUSTOMER_ID>2</CUSTOMER_ID> <CUSTOMER_NAME>Joe User</CUSTOMER_NAME> </XMLResultRowTag> </XMLResultRootTag></pre> <ul style="list-style-type: none"> ◆ INSERT – Executes an insert query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above. ◆ UPDATE – Executes an update query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above. ◆ ACTION – Executes an INSERT, UPDATE, or DELETE query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above. ◆ Stored procedure/function – Applies to stored procedures and functions.

Field	Description
	<p>Notes for stored procedures and functions:</p> <ul style="list-style-type: none"> ◆ When defining a Lightweight JDBC adapter instance using the GPM, you must declare a Query type of Stored Procedure/Function. When defining a Lightweight JDBC adapter instance using the BPML, you must declare a Query type of PROCEDURE. ◆ You should specify the stored procedure on the SQL parameter in the Lightweight JDBC adapter instance definition using the JDBC syntax. For example: <pre>{call StoredProcedureName(?, ?, ?)}</pre> <p>where ?,?,? are the arguments passed in from the parameter specifications (param1-param20) defined in the Lightweight JDBC adapter instance definition.</p> ◆ You should specify the function on the SQL parameter in the Lightweight JDBC adapter instance definition using the JDBC syntax. For example: <pre>{? = call product_count() }</pre> <p>where ? is the argument returned from the function. You must specify a parameter type that matches the format of the value returned from the function. For example: <pre><assign to="paramtype1">Integer</assign></pre> </p> ◆ There are limitations when using Oracle to call stored procedures. For information about these limitations, see <i>Calling Oracle Stored Procedures</i> on page 869.
SQL Statement (sql)	<p>Hard-coded SQL query that queries a database. It must be in valid SQL syntax.</p> <p>Note: You can specify the SQL query either using this field or when you add the Lightweight JDBC adapter instance in a business process.</p>

GPM Configuration

The following screen shows the graphical view of the GPM parameters for the Lightweight JDBC adapter. The inactive values were specified using the Lightweight JDBC adapter configuration. The active fields are

fields that cannot be configured in the Application or that are being overridden. There are no fields that must be configured on the Message From Service tab.

Name: LWJDBC - QueryForCustomerID
 Config: ExampleLWJDBCBusinessProcess

Message To Service | **Message From Service**

Output Msg: Obtain Message first, then Process Data
 Message Name: LightweightJDBCAdapterTypeInputMessage

Name	Value	Use XPath?
InitialWorkflowId		<input type="checkbox"/>
param1	number(ProcessData/CustomerInformation/ResultsFromQuery/Customer(number(NumberOfRecords))/customer_id/text())	<input checked="" type="checkbox"/>
param10		<input type="checkbox"/>
param11		<input type="checkbox"/>
param12		<input type="checkbox"/>
param13		<input type="checkbox"/>
param14		<input type="checkbox"/>
param15		<input type="checkbox"/>
param16		<input type="checkbox"/>
param17		<input type="checkbox"/>
param18		<input type="checkbox"/>
param19		<input type="checkbox"/>
param2		<input type="checkbox"/>
param20		<input type="checkbox"/>
param3		<input type="checkbox"/>
param4		<input type="checkbox"/>
param5		<input type="checkbox"/>
param6		<input type="checkbox"/>
param7		<input type="checkbox"/>
param8		<input type="checkbox"/>

Name	Value	Use XPath?
param9		<input type="checkbox"/>
paramtype1	Integer	<input type="checkbox"/>
paramtype10		<input type="checkbox"/>
paramtype11		<input type="checkbox"/>
paramtype12		<input type="checkbox"/>
paramtype13		<input type="checkbox"/>
paramtype14		<input type="checkbox"/>
paramtype15		<input type="checkbox"/>
paramtype16		<input type="checkbox"/>
paramtype17		<input type="checkbox"/>
paramtype18		<input type="checkbox"/>
paramtype19		<input type="checkbox"/>
paramtype2		<input type="checkbox"/>
paramtype20		<input type="checkbox"/>
paramtype3		<input type="checkbox"/>
paramtype4		<input type="checkbox"/>
paramtype5		<input type="checkbox"/>
paramtype6		<input type="checkbox"/>
paramtype7		<input type="checkbox"/>
paramtype8		<input type="checkbox"/>

paramtype9		<input type="checkbox"/>
pool	mysqlTrainingPool	<input type="checkbox"/>
query_type	Select	<input type="checkbox"/>
result_name	RootResultXmlElement	<input type="checkbox"/>
row_name	ResultsFromSQL	<input type="checkbox"/>
sql	SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2...	<input type="checkbox"/>
StartNewWorkflow	This Lightweight JDBC Adapter will not start a new busine...	<input type="checkbox"/>

The following example shows the corresponding BPML parameters for the Lightweight JDBC adapter GPM parameters. This example business process takes the primary document, which contains the query results, and writes it to process data.

```
<process name="ExampleLWJDBC">
  <operation name="LWJDBC - QueryForCustomerID">
    <participant name="ExampleLWJDBCBusinessProcess" />
    <output message="LightweightJDBCAdapterTypeInputMessage">
```

```

<assign to="param1"
from="number (/ProcessData/CustomerInformation/ResultsFromQuery/Customer[number (/NumberOfRecords)]/customer_id/text())" />
<assign to="paramtype1">Integer</assign>
<assign to="query_type">SELECT</assign>
<assign to="sql">SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2 WHERE CUSTOMER_ID =
?</assign>
<assign to="." from="*" />
</output>
<input message="inmsg">
<assign to="NumberOfCustomers"
from="DocToDOM(PrimaryDocument)/ResultsFromSQL/CUSTOMERS" />
</input>
</operation>
</sequence>
</process>

```

The following table describes the fields used to configure the Lightweight JDBC adapter in the GPM. This table contains only the fields that are configured in the GPM. See *The Application Configuration* on page 855 for parameters you can specify in the GPM or in the Application configuration.

Field	Description
Config	Name of the service configuration.
param1 - param20	Specifies the values that are passed to the SQL statement that is to be executed against the database. You can either specify constant values or use XPath query statements to retrieve information from process data to pass to the SQL statement. If you use XPath query statements, you must also select the "Use XPath?" check box in the Service Editor of the GPM for the parameter. See the <i>Lightweight JDBC Business Process Usage</i> on page 864 for an example of using Xpath to set these values. Optional.
paramtype1 - paramtype20	Parameter type that corresponds to the parameter number (for example, param1 and paramtype1). Every parameter specified must have a corresponding parameter type. Valid values: <ul style="list-style-type: none"> ◆ Cursor ◆ String ◆ CharacterStream ◆ CharacterStreamFromDocument ◆ Integer ◆ Long ◆ Double ◆ Float ◆ Date

Large Text Data Support

Two values are available for large text data (clob) support:

CharacterStream is used to insert or update data to the database that is in process data.

CharacterStreamFromDocument is used to insert or update data to the database from the primary document.

To insert data to the database or to update the database using the **CharacterStream** or **CharacterStreamFromDocument** values, there are two new query types: **INSERT** and **UPDATE**.

There is also a parameter that you can add to your business process manually (this parameter is not available through the GPM): **write_characterstream_to_document**. Valid values for this parameter are **YES** and **NO**. This parameter enables you to place the contents of large data inside of a document during a **SELECT** query, because **XML** has size limitations on pcddata. The primary document will still contain the result tree but the name of the document will replace the large data. This new parameter is optional and defaults to **NO** for backward compatibility.

Note: When inserting, updating, or selecting clobs from Oracle, ensure the **varDataClassName** in the **jdbc.properties.in** file is pointing to **OracleBlobVarData** for an external pool.

The Lightweight JDBC adapter will attempt to get a connection a set number of times. The number of retries is configurable by changing the **externalDBPoolRetries** property in the **noapp.properties.in** file.

The following example illustrates using the parameter to write the contents of the data returned from the database directly to a document:

```
<operation name="LightweightJDBCAdapterType">
  <participant name="LightweightJDBCAdapterQuery" />
  <output message="LightweightJDBCAdapterTypeInputMessage">
    <assign to="result_name">result</assign>
    <assign to="sql">SELECT ID, XML_MESSAGE FROM CUSTOMER WHERE
CUSTOMER_ID=?</assign>
    <assign to="pool">mysqlPool_local</assign>
    <assign to="param1" from="customer_id/text()" />
    <assign to="paramtype1">String</assign>
    <assign to="write_characterstream_to_document">Yes</assign>
    <assign to="row_name">row</assign>
    <assign to="query_type">SELECT</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

The following example illustrates a sample of what you would see in the primary document:

```
<?xml version="1.0" encoding="UTF-8"?>
  <result>
    <row>
      <ID>1</ID>
      <XML_MESSAGE>doc-1
```

The parameter types are used by the Lightweight JDBC adapter to insert or update large text data into the database. The insert and update statements differ in how they are implemented, so you must specify the INSERT or UPDATE action type. Several examples follow.

Example 1

Updates the column with the contents of the primary document. In this example, the parameter type is `CharacterStreamFromDocument`, so the parameter value is not required; instead, the value is taken from the primary document.

Example 2

Updates a column with the string returned from an Xpath query against process data. In this example, the `CharacterStream` parameter type is used, so the parameter value is required.

Example 3

Inserts a row with the string returned from an Xpath query against process data.

Example 4

Writes the contents of the data returned from a database directly to a document (note that this is not the primary document). The primary document will still contain the XML result tree; however, the data in the document is replaced by a document name.

This is an example of what you should expect to see in the primary document:

This is an Oracle example for updating a clob column. The first parameter is used to find the row in the table. The second parameter will be used to update the value.

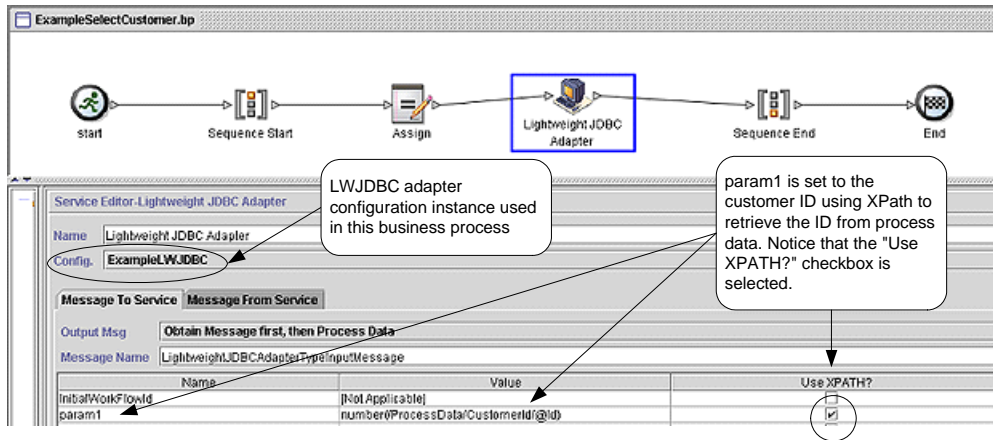
Lightweight JDBC Business Process Usage

This section contains additional examples using the Lightweight JDBC adapter, including sample configurations for using Oracle stored procedures and functions.

The following sample document is the primary document passed to the business process for the SELECT and ACTION query type examples.

SELECT Query Type Example

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to SELECT customer information based on the customer ID passed into the business process.

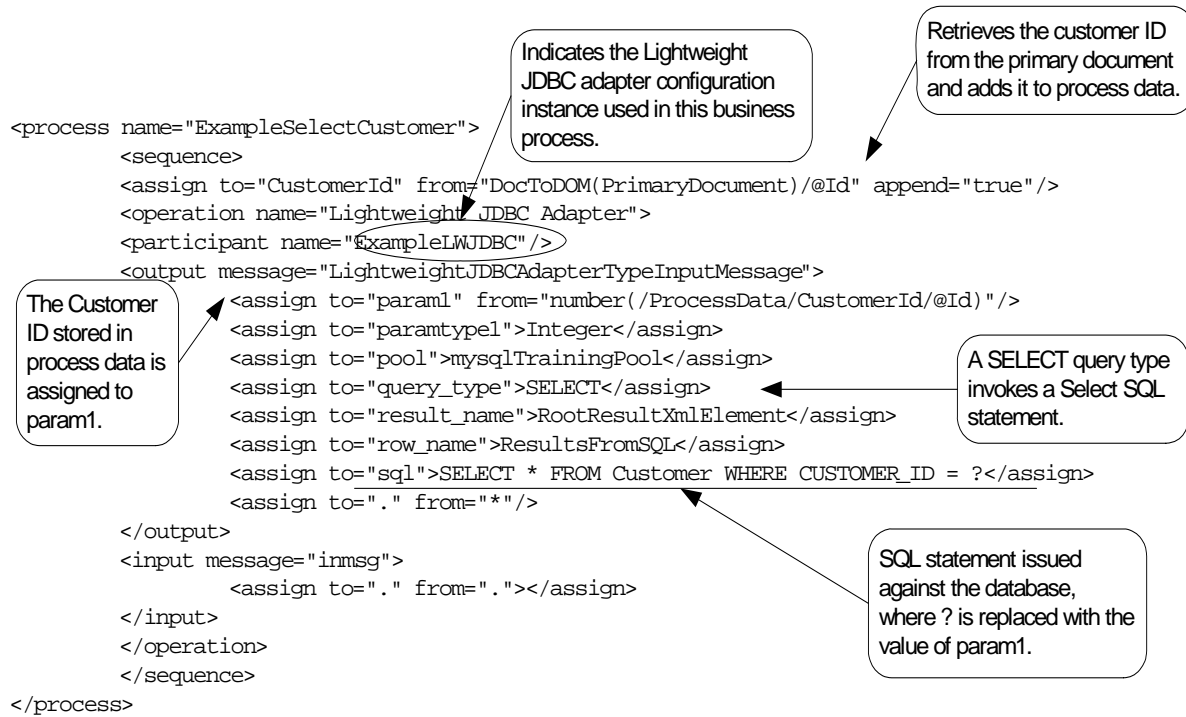


The Select query type invokes a Select SQL statement.

sql is set to the actual SQL statement issued against the database, where "?" is replaced with the value of param1.

pool	mysqlTrainingPool	
query_type	Select	<input type="checkbox"/>
result_name	RootResultXmlElement	<input type="checkbox"/>
row_name	ResultsFromSQL	<input type="checkbox"/>
sql	SELECT * FROM Customer WHERE CUSTOMER_ID = ?	<input type="checkbox"/>
StartNewWorkflow	This Lightweight JDBC Adapter will not start a new busine...	<input type="checkbox"/>

The following example illustrates the same business process using BPML.



Message to the Lightweight JDBC Adapter Instance

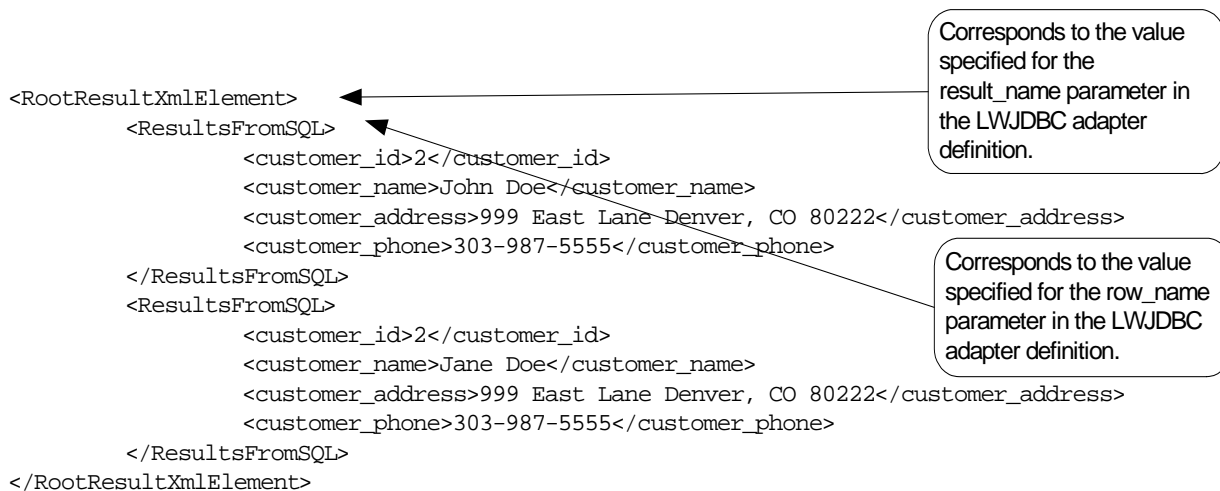
The following example shows the message sent to the Lightweight JDBC adapter when you run the business process. The message uses the parameters defined within the `<output>` tags of the Lightweight JDBC adapter definition in the previous BPML example, along with the parameters defined during configuration of the Lightweight JDBC adapter instance.

Note: You can view the message sent to the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message To Service”.

Message from the Lightweight JDBC Adapter Instance

The following example shows the table definition used to query against the Lightweight JDBC adapter. In the previous BPML example, the assign statement within the <input> tags of the Lightweight JDBC adapter definition (<assign to="." from="."></assign>) indicates that the results are returned as a primary document.

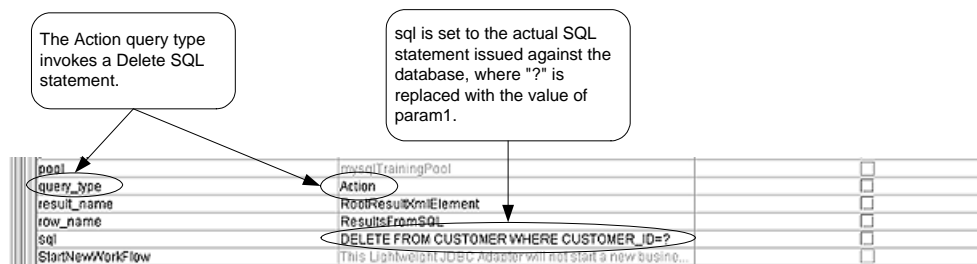
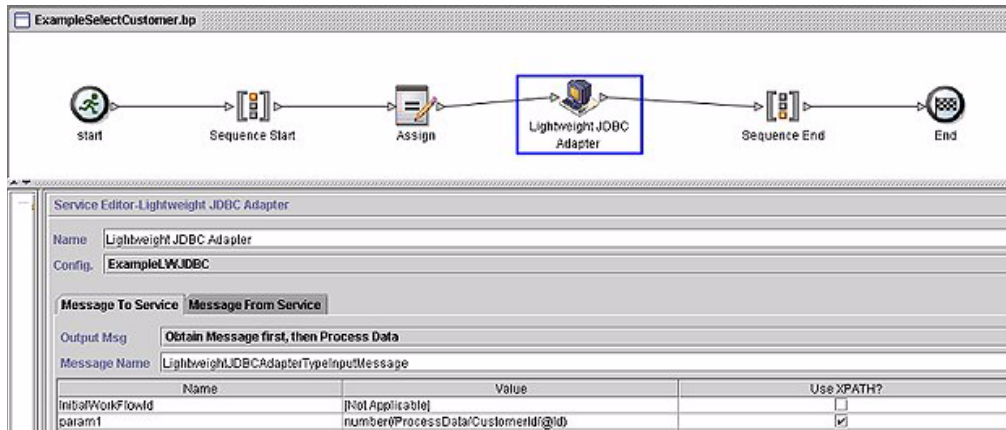
Notice how the columns from the Customer table become XML elements in the results returned from the Lightweight JDBC adapter, as shown below in the message returned from the Lightweight JDBC adapter.



Note: You can view the message returned from the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message From Service”.

ACTION Query Type Example

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to delete customer information from a database.



The following example illustrates the same business process using BPML.

```
<process name="ExampleDeleteCustomer">
  <sequence>
    <assign to="CustomerId" from="DocToDOM(PrimaryDocument)/@Id" append="true"/>
    <operation name="Lightweight JDBC Adapter">
      <participant name="ExampleLWJDBC"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="number(/ProcessData/CustomerId/@Id)"/>
        <assign to="paramtype1">Integer</assign>
        <assign to="query_type">ACTION</assign>
        <assign to="result_name">RootResultXmlElement</assign>
        <assign to="row_name">ResultsFromSQL</assign>
        <assign to="sql">DELETE FROM CUSTOMER WHERE CUSTOMER_ID=?</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="." />
      </input>
    </operation>
  </sequence>
</process>
```

The ACTION query type invokes a Delete SQL statement in this example.

Actual SQL statement issued against the database, where "?" is replaced with the value of param1.

Message to the Lightweight JDBC Adapter Instance

The following example shows the message sent to the Lightweight JDBC adapter when you run the business process. The message uses the parameters defined within the <output> tags of the Lightweight JDBC adapter definition in the previous BPML, along with the parameters defined for the Lightweight JDBC adapter instance during configuration.

Note: You can view the message sent to the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message To Service”.

Message from the Lightweight JDBC Adapter Instance

The following example shows the message returned from the Lightweight JDBC adapter to the business process. The assign statement within the <input> tags of the Lightweight JDBC adapter definition (<assign to="." from="."></assign>) indicates that the results are returned as a primary document.

```
<RootResultXmlElement>
<ResultsFromSQL>
<Rows_Affected>1</Rows_Affected>
</ResultsFromSQL>
</RootResultXmlElement>
```

The Lightweight JDBC adapter returns the number of rows deleted.

Note: You can view the message returned from the Lightweight JDBC adapter from the Business Process Monitor by selecting “Instance Data” at the step where the adapter instance is run and then selecting “Message From Service”.

Calling Oracle Stored Procedures

This section describes limitations when using the Lightweight JDBC adapter to call Oracle stored procedures. These limitations apply only to Oracle databases.

The Lightweight JDBC adapter processes only one result set if you script multiple queries to run in stored procedures.

The first argument in an Oracle stored procedure must be declared as an output parameter, and the stored procedure must return a value to this parameter even if the value is not needed in the business process.

You must specify a value for the first parameter (param1) in the Lightweight JDBC adapter instance to correspond with the first argument of the stored procedure, even if the parameter is not used in the stored procedure.

Stored Procedure Example

The following example shows a database script for creating an Oracle stored procedure. The stored procedure updates the account status for a customer. Notice that the first argument of the stored procedure is declared as an output parameter and is set to a number, even though this value will not be used in the business process.

```
    v_dummy := '4';  
END;  
/
```

The following sample document is the primary document passed into the example business process used to call the stored procedure.

```
<PurchaseOrder>  
  <Account Id="100">  
    <Customer Id="2"/>  
    <Status>ACTIVE</Status>  
  </Account>  
</PurchaseOrder>
```

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to start the stored procedure.

ExampleUpdateAccountStatus.bp

Service Editor - Lightweight JDBC Adapter

Name: Lightweight JDBC Adapter

Config: ExampleLWJDBCBusinessProcess

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: LightweightJDBCAdapterTypeInputMessage

Name	Value	Use XPath?
IndiaWorkFlowId		<input type="checkbox"/>
param1	1	<input type="checkbox"/>
param1 0		<input type="checkbox"/>
param1 1		<input type="checkbox"/>
param1 2		<input type="checkbox"/>
param1 3		<input type="checkbox"/>
param1 4		<input type="checkbox"/>
param1 5		<input type="checkbox"/>
param1 6		<input type="checkbox"/>

The value set for param1 is not used in the stored procedure.

param2	number(/ProcessData/CustomerId/@id)	<input checked="" type="checkbox"/>
param3	number(/ProcessData/Account/@id)	<input checked="" type="checkbox"/>
param4	/ProcessData/AccountStatus/text()	<input checked="" type="checkbox"/>
param5		<input type="checkbox"/>
param6		<input type="checkbox"/>
param7		<input type="checkbox"/>
param8		<input type="checkbox"/>
param9		<input type="checkbox"/>
paramtype1	Integer	<input type="checkbox"/>
paramtype1 0		<input type="checkbox"/>
paramtype1 1		<input type="checkbox"/>
paramtype1 2		<input type="checkbox"/>
paramtype1 3		<input type="checkbox"/>
paramtype1 4		<input type="checkbox"/>
paramtype1 5		<input type="checkbox"/>
paramtype1 6		<input type="checkbox"/>

param2 is set to the customer Id.
param3 is set to the customer account Id.
param4 is set to the new account status.
These values are used in the stored procedure to update the customer account status.

paramtype17		<input type="checkbox"/>
paramtype18		<input type="checkbox"/>
paramtype19		<input type="checkbox"/>
paramtype2	Integer	<input type="checkbox"/>
paramtype20		<input type="checkbox"/>
paramtype3	Integer	<input type="checkbox"/>
paramtype4	String	<input type="checkbox"/>
paramtype5		<input type="checkbox"/>
paramtype6		<input type="checkbox"/>
paramtype7		<input type="checkbox"/>
paramtype8		<input type="checkbox"/>
paramtype9		<input type="checkbox"/>
pool		<input type="checkbox"/>
query_type	PROCEDURE	<input type="checkbox"/>
result_name	RootResultXmlElement	<input type="checkbox"/>
row_name	ResultsFromSQL	<input type="checkbox"/>
sql	{call P_Update_Account(?,?,?)}	<input type="checkbox"/>
StartNewWorkFlow	NO	<input type="checkbox"/>

The PROCEDURE query type invokes a stored procedure in this example.

sql is set to the stored procedure to invoke, where "?,?,?" is replaced with the values defined for param1, param2, param3, and param4.

The following example illustrates the same business process using BPML:

```

<process name="ExampleUpdateAccountStatus">
  <sequence>
    <assign to="CustomerId" from="DocToDOM(PrimaryDocument)/Account/Customer/@Id"
append="true"/>
    <assign to="Account" from="DocToDOM(PrimaryDocument)/Account/@Id" append="true"/>
    <assign to="AccountStatus" from="DocToDOM(PrimaryDocument)/Account/Status/text()"
append="true"/>
    <operation name="Lightweight JDBC Adapter">
      <participant name="ExampleLWJDBCBusinessProcess"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1">1</assign>
        <assign to="paramtype1">Integer</assign>
        <assign to="param2" from="number(/ProcessData/CustomerId/@Id)"/>
        <assign to="paramtype2">Integer</assign>
        <assign to="param3" from="number(/ProcessData/Account/@Id)"/>
        <assign to="paramtype3">Integer</assign>
        <assign to="param4" from="/ProcessData/AccountStatus/text()"/>
        <assign to="paramtype4">String</assign>
        <assign to="pool">oraclePool</assign>
        <assign to="query_type">PROCEDURE</assign>
        <assign to="result_name">RootResultXmlElement</assign>
        <assign to="row_name">ResultsFromSQL</assign>
        <assign to="sql">{call P_Update_Account(?,?,?,?)}</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="." />
      </input>
    </operation>
  </sequence>
</process>

```

Specify PROCEDURE as the query type when using BPML.

Notice the format used to call stored procedures.

Calling Oracle Functions

The following example shows a database script for creating a simple Oracle function. The function retrieves and returns the number of products in the database:

```

CREATE OR REPLACE FUNCTION product_count
RETURN number
IS
  v_product_count number;

BEGIN
  SELECT count(*) INTO v_product_count
  FROM product;

RETURN v_product_count;
END;
/

```


The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to start the function:

ExampleGetProductCount.bp

Service Editor - Lightweight JDBC Adapter

Name: Lightweight JDBC Adapter
 Config: ExampleLVJDBCBusinessProcess

Message To Service: Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: LightweightJDBCAdapterTypeInputMessage

Name	Value	Use XPath?
paramtype1	Integer	<input type="checkbox"/>
paramtype10		<input type="checkbox"/>
paramtype11		<input type="checkbox"/>
paramtype12		<input type="checkbox"/>
paramtype13		<input type="checkbox"/>
paramtype14		<input type="checkbox"/>
paramtype15		<input type="checkbox"/>
paramtype16		<input type="checkbox"/>
paramtype17		<input type="checkbox"/>
paramtype18		<input type="checkbox"/>
paramtype19		<input type="checkbox"/>

Advanced

No input parameters are specified for the function. However, you must specify a parameter type that matches the format of the value returned from the function. In this case, we set paramtype1 to Integer because the function returns the number of products.

paramtype19		<input type="checkbox"/>
paramtype2		<input type="checkbox"/>
paramtype20		<input type="checkbox"/>
paramtype3		<input type="checkbox"/>
paramtype4		<input type="checkbox"/>
paramtype5		<input type="checkbox"/>
paramtype6		<input type="checkbox"/>
paramtype7		<input type="checkbox"/>
paramtype8		<input type="checkbox"/>
paramtype9		<input type="checkbox"/>
pool	oraclePool	<input type="checkbox"/>
query_type	Stores procedure/function	<input type="checkbox"/>
result_name	TestIDBLookupResult	<input type="checkbox"/>
row_name	ResultsFromSQL	<input type="checkbox"/>
sql	{? = call product_count(?)}	<input type="checkbox"/>
StartNewWorkflow	This Lightweight JDBC Adapter will not start a new busine...	<input type="checkbox"/>

sql is set to the function to invoke, where "?" is replaced with the value returned from the function and mapped to paramtype1.

The following example illustrates the same business process using BPML:

```
<process name="ExampleGetProductCount">
  <sequence>
    <operation name="Lightweight JDBC Adapter">
      <participant name="ExampleLWJDBCBusinessProcess"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="paramtype1">Integer</assign>
        <assign to="pool">oraclePool</assign>
        <assign to="query_type">PROCEDURE</assign>
        <assign to="result_name">TestDBLookupResult</assign>
        <assign to="row_name">ResultsFromSQL</assign>
        <assign to="sql">{? = call product_count()}</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="." />
      </input>
    </operation>
  </sequence>
</process>
```

Specify PROCEDURE as the query type when using BPML.

Notice the format used to call functions.

Lock Service (Build 4300 - Build 4317)

The Lock service enables a business process to request, renew, or delete a lock for a particular resource. The following table provides an overview of the Lock service:

System Name	LockService
Graphical Process Modeler (GPM) category	All Services
Description	Enables a business process to request, renew, or delete a lock for a particular resource.
Business usage	If you have a business process, or resources within a business process, that should never have more than one instance running at a time, you can use the Lock service to prevent other instances of the business process, or just a certain part of the business process, from running until the lock is released.
Usage example	You have a business process that uses a configuration of the File System adapter called "Inbound Invoices." To prevent this particular configuration of the File System adapter from being invoked by another business process while processing data in the current business process, you add the Lock service before the File System adapter in the business process, to lock that resource. You add a second instance of the Lock service after the File System adapter in the business process to release the lock once the File System adapter processing is complete.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	As part of a business process.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	None
Testing considerations	Use the <i>Business Process Example</i> on page 877 as a test.

How the Lock Service Works

The Lock service secures a business process and prevents other business processes from using the locked resources until the lock is released. The Lock service uses the lock key and duration time that you set in the GPM for the business process to identify the lock to set and how long to keep the resources locked. You can

lock all of the activities and services used in a business process by adding the Lock service to the beginning of a business process, after the Start and Sequence Start activities. Or, you can lock just some of the activities or services in a business process by adding the Lock service directly before the activities or services to be locked.

To release the lock, add another instance of the Lock service to the business process directly after the group of locked activities and services. If the entire business process was locked, add the second Lock service to the end of the business process, directly before the End Sequence and End activities.

Implementing the Lock Service

To implement the Lock service, complete the following tasks:

1. Create a configuration of the Lock service, or use the configuration installed with the Application, LockService. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Lock Service* on page 876.
2. Include two instances of the Lock service in your business process (one to lock resources, another to unlock them).
3. Specify field settings for each instance in the GPM as necessary. Ensure that you set the first instance to use the Lock action, and the second to use the Unlock action. Also ensure that you specify the same lock key for both.

Configuring the Lock Service

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
ACTION	<p>Action performed for the requested lock. Valid values are:</p> <ul style="list-style-type: none"> ◆ Check – Looks for the existence of a lock that is identified in the LOCK_KEY field. If the lock exists, the LOCK_EXIST output workflow parameter is set to true. If the lock does not exist, LOCK_EXIST is set to false. ◆ Lock – Create (default) ◆ Touch – Renew ◆ Unlock – Delete <p>Optional.</p> <p>Note: As a best practice, always use the Lock service in pairs in your business processes—one instance to lock the business process, and one to unlock the business process after the necessary operations have completed.</p>
DURATION	<p>Time, in milliseconds, that the lock is applied. Required. The lock will time out or expire after this time.</p> <p>Note: If a business process halts due to an error, the service configuration will remain locked until you manually release the lock or restart the Application.</p>

Field	Description
LOCK_KEY	The key for obtaining the lock. Required. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.
USER	User name associated with the lock (informational only). Optional. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.

Parameter(s) That Must be Added in BPML

The following additional parameter(s) can be used with Lock service by editing the BPML:

Parameter	Description
CLEAR_ON_START_UP	Clears the lock after you restart the Application. Use when you want to ensure that the lock is always clean when the Application is restarted. Optional. Valid values are true and false.

Business Process Example

The following example illustrates how the Lock service could be used in a business process.

In this business process, a lock is applied by the Lock service. The lock is set to a duration of 600,000 milliseconds, and uses “Lock1” as the lock key. In this business process, the resource being locked is an instance of the Sleep service. Note that the sleep interval (duration) is 45 seconds. The Sleep service is followed by a second instance of the Lock service that releases the lock, thereby freeing up this configuration of the Sleep service configuration for other processes.

Note: The business process also includes onFault activities and messages in the event that one of the lock activities fails.

```
<process name="LockExample">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="DURATION">600000</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="CLEAR_ON_START_UP">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <sequence name="Process">
      <operation name="Sleep">
        <participant name="TestSleepService"/>
        <output message="Xout">
          <assign to="SLEEP_INTERVAL">45</assign>
        </output>
      </operation>
    </sequence>
  </sequence>
</process>
```

```

    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

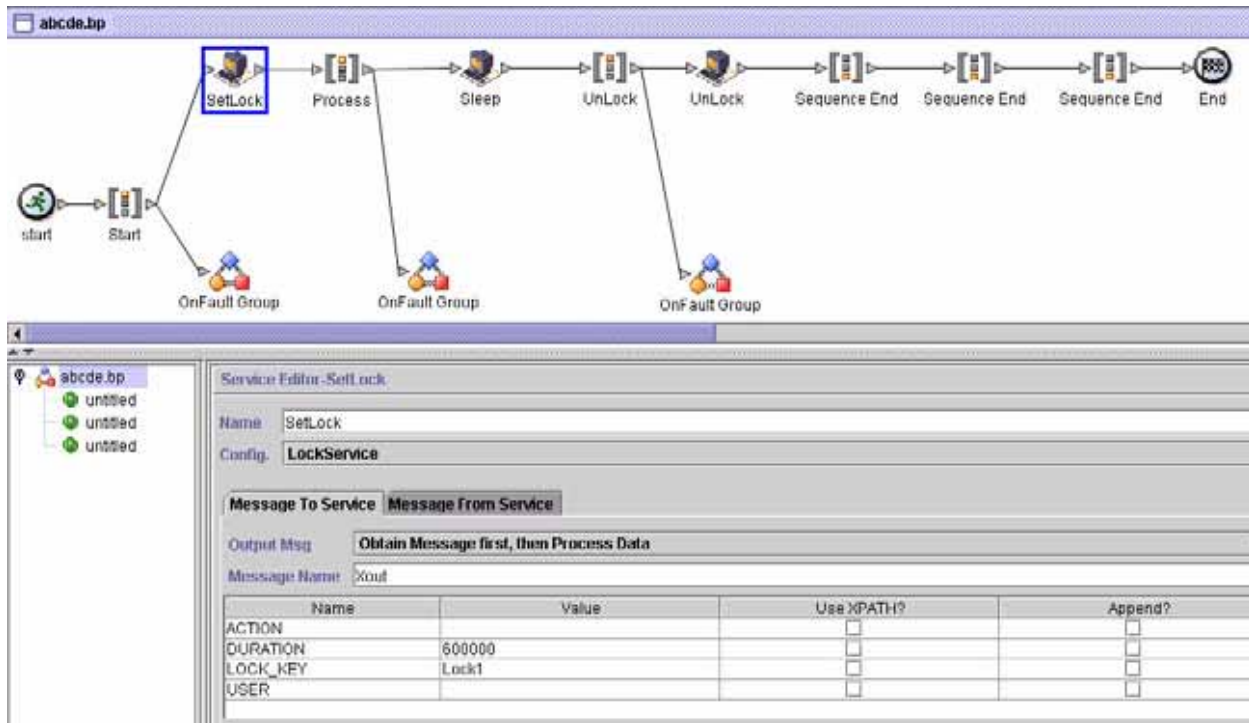
<sequence name="UnLock">
  <operation name="UnLock">
    <participant name="LockService"/>
    <output message="Xout">
      <assign to="ACTION">unlock</assign>
      <assign to="LOCK_KEY">Lock1</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <onFault>
    <assign to="UnLock_Msg" append="true">Failed to obtain an unlock!</assign>
  </onFault>
</sequence>
<onFault>
  <operation>
    <participant name="LockService"/>
    <output message="Xout">
      <assign to="ACTION">unlock</assign>
      <assign to="LOCK_KEY">Lock1</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  </onFault>
</sequence>
<onFault>
  <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
</onFault>
</sequence>
</process>

```

The following GPM screen shows the example business process graphically. Note that the instance of the Lock service called SetLock is selected and its properties are displayed in the Service Editor in the lower half of the screen:



Lock Service – Frequently Asked Questions

How do I determine what the duration of a lock should be?

Set the lock duration value carefully (generally 2-3 times what the estimated time of execution will be) so that lock does not time out before the business process reaches the unlock step.

If the lock or unlock step fails (or any step in the business process between the two Lock services), are there any “cleanup” activities that need to be done?

Halted business processes can be terminated, or they can be left in halted state until the issue is resolved, and then restarted or resumed (as appropriate for the persistence level). The path to take depends on the needs of your business with regard to the business process itself. For example, is it mission-critical? Do other processes depend on its completion? Is the locked resource going to cause other business processes that use it to halt? Was the error caused by a problem in the configuration of the locked resource? Will this need to be corrected before using it again?

There are two general steps to follow first:

1. Check the lock manager page to see what’s locked.
2. Check **Business Process > BP Monitor > Current Processes** for more information about the error.

Then, once you have determined what caused the error, you can decide when to release the lock and when to terminate, restart, or resume the business process.

To manually release a locked resource:

Go to **Operations > Lock Manager**, and click **Go!** in the List panel. The locked resources are displayed on a results page. Locate the resources from your business process that are locked and clear the Lock checkbox for the resources.

To terminate, restart, or resume a business process:

Go to **Business Process > Monitor > Current Processes** and select the ID of the halted instance of your business process. From the page displayed, you can select the appropriate action for this business process: terminate, restart, or resume.

Are there any best practices for using the Lock service in a business process?

Use the Lock service twice in a business process—one to lock resources and one to unlock them. The first instance precedes the resources to be locked and the second instance follows them. See the *Business Process Example* on page 877 for a graphical representation. Do not use just one instance of the Lock service in a business process and let it expire instead of using a second Lock service to release the lock.

Lock Service (Build 4318 - Build 4321)

The Lock service enables a business process to request, renew, or delete a lock for a particular resource. The following table provides an overview of the Lock service:

System Name	LockService
Graphical Process Modeler (GPM) category	All Services
Description	Enables a business process to request, renew, or delete a lock for a particular resource.
Business usage	If you have a business process, or resources within a business process, that should never have more than one instance running at a time, you can use the Lock service to prevent other instances of the business process, or just a certain part of the business process, from running until the lock is released.
Usage example	You have a business process that uses a configuration of the File System adapter called "Inbound Invoices." To prevent this particular configuration of the File System adapter from being invoked by another business process while processing data in the current business process, you add the Lock service before the File System adapter in the business process, to lock that resource. You add a second instance of the Lock service after the File System adapter in the business process to release the lock once the File System adapter processing is complete.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	As part of a business process.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	None
Testing considerations	Use the <i>Business Process Example</i> on page 883 as a test.

How the Lock Service Works

The Lock service secures a business process and prevents other business processes from using the locked resources until the lock is released. The Lock service uses the lock key and duration time that you set in the GPM for the business process to identify the lock to set and how long to keep the resources locked. You can

lock all of the activities and services used in a business process by adding the Lock service to the beginning of a business process, after the Start and Sequence Start activities. Or, you can lock just some of the activities or services in a business process by adding the Lock service directly before the activities or services to be locked.

To release the lock, add another instance of the Lock service to the business process directly after the group of locked activities and services. If the entire business process was locked, add the second Lock service to the end of the business process, directly before the End Sequence and End activities.

Implementing the Lock Service

To implement the Lock service, complete the following tasks:

1. Create a configuration of the Lock service, or use the configuration installed with the Application, LockService. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Lock Service* on page 882.
2. Include two instances of the Lock service in your business process (one to lock resources, another to unlock them).
3. Specify field settings for each instance in the GPM as necessary. Ensure that you set the first instance to use the Lock action, and the second to use the Unlock action. Also ensure that you specify the same lock key for both.

Configuring the Lock Service

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
ACTION	<p>Action performed for the requested lock. Valid values are:</p> <ul style="list-style-type: none"> ◆ Check – Looks for the existence of a lock that is identified in the LOCK_KEY field. If the lock exists, the LOCK_EXIST output workflow parameter is set to true. If the lock does not exist, LOCK_EXIST is set to false. ◆ List Locks - Returns a document containing an XML list of the details of the currently active locks. ◆ Lock – Create (default) ◆ Touch – Renew ◆ Unlock – Delete <p>Optional.</p> <p>Note: As a best practice, always use the Lock service in pairs in your business processes—one instance to lock the business process, and one to unlock the business process after the necessary operations have completed.</p>

Field	Description
DURATION	Time, in milliseconds, that the lock is applied. Required. The lock will time out or expire after this time. Note: If a business process halts due to an error, the service configuration will remain locked until you manually release the lock or restart the Application.
LOCK_KEY	The key for obtaining the lock. Required. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.
MAX_MEM_LOCK_PER_NODE_LISTED	Maximum number of local locks returned from each node. Valid value is any number greater than 0. Use this parameter if the value you specified in the ACTION field is List Locks. The default value is the value you specified in the defaultMaxMemLocksListedPerNode property in the centralops.properties.in file. Optional.
USER	User name associated with the lock (informational only). Optional. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.

Parameter(s) That Must be Added in BPML

The following additional parameter(s) can be used with Lock service by editing the BPML:

Parameter	Description
CLEAR_ON_START_UP	Clears the lock after you restart the Application. Use when you want to ensure that the lock is always clean when the Application is restarted. Optional. Valid values are true and false.

Business Process Example

The following example illustrates how the Lock service could be used in a business process.

In this business process, a lock is applied by the Lock service. The lock is set to a duration of 600,000 milliseconds, and uses “Lock1” as the lock key. In this business process, the resource being locked is an instance of the Sleep service. Note that the sleep interval (duration) is 45 seconds. The Sleep service is followed by a second instance of the Lock service that releases the lock, thereby freeing up this configuration of the Sleep service configuration for other processes.

Note: The business process also includes onFault activities and messages in the event that one of the lock activities fails.

```
<process name="LockExample">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="DURATION">600000</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="CLEAR_ON_START_UP">>true</assign>
        <assign to="." from="*"></assign>
      </output message="Xout">
    </operation name="SetLock">
  </sequence name="Start">
</process name="LockExample">
```

```

    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <sequence name="Process">
    <operation name="Sleep">
      <participant name="TestSleepService"/>
      <output message="Xout">
        <assign to="SLEEP_INTERVAL">45</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

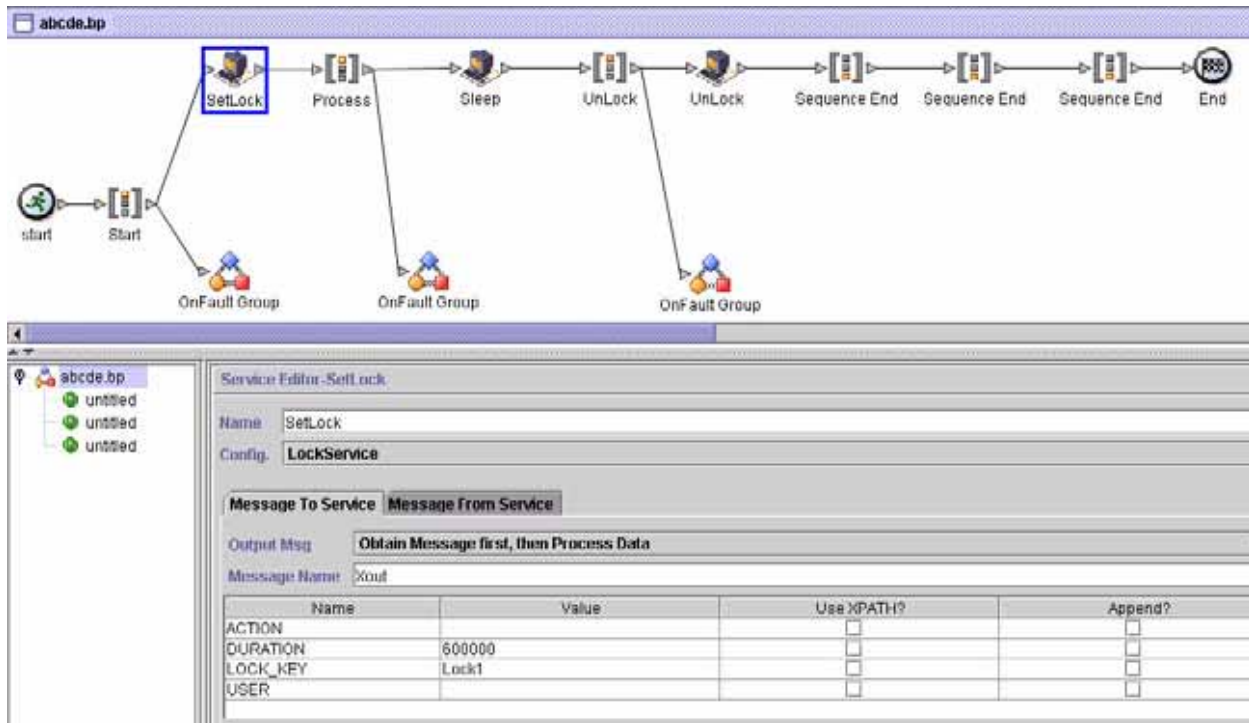
    <sequence name="UnLock">
      <operation name="UnLock">
        <participant name="LockService"/>
        <output message="Xout">
          <assign to="ACTION">unlock</assign>
          <assign to="LOCK_KEY">Lock1</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="Xin">
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <onFault>
        <assign to="UnLock_Msg" append="true">Failed to obtain an unlock!</assign>
      </onFault>
    </sequence>
    <onFault>
      <operation>
        <participant name="LockService"/>
        <output message="Xout">
          <assign to="ACTION">unlock</assign>
          <assign to="LOCK_KEY">Lock1</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="Xin">
          <assign to="." from="*"></assign>
        </input>
      </operation>

    </onFault>
  </sequence>
  <onFault>
    <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
  </onFault>
</sequence>
</process>

```

The following GPM screen shows the example business process graphically. Note that the instance of the Lock service called SetLock is selected and its properties are displayed in the Service Editor in the lower half of the screen:



Lock Service – Frequently Asked Questions

How do I determine what the duration of a lock should be?

Set the lock duration value carefully (generally 2-3 times what the estimated time of execution will be) so that lock does not time out before the business process reaches the unlock step.

If the lock or unlock step fails (or any step in the business process between the two Lock services), are there any “cleanup” activities that need to be done?

Halted business processes can be terminated, or they can be left in halted state until the issue is resolved, and then restarted or resumed (as appropriate for the persistence level). The path to take depends on the needs of your business with regard to the business process itself. For example, is it mission-critical? Do other processes depend on its completion? Is the locked resource going to cause other business processes that use it to halt? Was the error caused by a problem in the configuration of the locked resource? Will this need to be corrected before using it again?

There are two general steps to follow first:

1. Check the lock manager page to see what’s locked.
2. Check **Business Process > BP Monitor > Current Processes** for more information about the error.

Then, once you have determined what caused the error, you can decide when to release the lock and when to terminate, restart, or resume the business process.

To manually release a locked resource:

Go to **Operations > Lock Manager**, and click **Go!** in the List panel. The locked resources are displayed on a results page. Locate the resources from your business process that are locked and clear the Lock checkbox for the resources.

To terminate, restart, or resume a business process:

Go to **Business Process > Monitor > Current Processes** and select the ID of the halted instance of your business process. From the page displayed, you can select the appropriate action for this business process: terminate, restart, or resume.

Are there any best practices for using the Lock service in a business process?

Use the Lock service twice in a business process—one to lock resources and one to unlock them. The first instance precedes the resources to be locked and the second instance follows them. See the *Business Process Example* on page 883 for a graphical representation. Do not use just one instance of the Lock service in a business process and let it expire instead of using a second Lock service to release the lock.

Lock Service (Build 4322 or higher)

The Lock service enables a business process to request, renew, or delete a lock for a particular resource. The following table provides an overview of the Lock service:

System Name	LockService
Graphical Process Modeler (GPM) category	All Services
Description	Enables a business process to request, renew, or delete a lock for a particular resource.
Business usage	If you have a business process, or resources within a business process, that should never have more than one instance running at a time, you can use the Lock service to prevent other instances of the business process, or just a certain part of the business process, from running until the lock is released.
Usage example	You have a business process that uses a configuration of the File System adapter called "Inbound Invoices." To prevent this particular configuration of the File System adapter from being invoked by another business process while processing data in the current business process, you add the Lock service before the File System adapter in the business process, to lock that resource. You add a second instance of the Lock service after the File System adapter in the business process to release the lock once the File System adapter processing is complete.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	As part of a business process.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	None
Testing considerations	Use the <i>Business Process Example</i> on page 889 as a test.

How the Lock Service Works

The Lock service secures a business process and prevents other business processes from using the locked resources until the lock is released. The Lock service uses the lock key and duration time that you set in the GPM for the business process to identify the lock to set and how long to keep the resources locked. You can

lock all of the activities and services used in a business process by adding the Lock service to the beginning of a business process, after the Start and Sequence Start activities. Or, you can lock just some of the activities or services in a business process by adding the Lock service directly before the activities or services to be locked.

To release the lock, add another instance of the Lock service to the business process directly after the group of locked activities and services. If the entire business process was locked, add the second Lock service to the end of the business process, directly before the End Sequence and End activities.

Implementing the Lock Service

To implement the Lock service, complete the following tasks:

1. Create a configuration of the Lock service, or use the configuration installed with the Application, LockService. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Lock Service* on page 888.
2. Include two instances of the Lock service in your business process (one to lock resources, another to unlock them).
3. Specify field settings for each instance in the GPM as necessary. Ensure that you set the first instance to use the Lock action, and the second to use the Unlock action. Also ensure that you specify the same lock key for both.

Configuring the Lock Service

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
ACTION	<p>Action performed for the requested lock. Valid values are:</p> <ul style="list-style-type: none"> ◆ Check – Looks for the existence of a lock that is identified in the LOCK_KEY field. If the lock exists, the LOCK_EXIST output workflow parameter is set to true. If the lock does not exist, LOCK_EXIST is set to false. ◆ List Locks - Returns a document containing an XML list of the details of the currently active locks. ◆ Lock – Create (default) ◆ Touch – Renew ◆ Unlock – Delete <p>Optional.</p> <p>Note: As a best practice, always use the Lock service in pairs in your business processes—one instance to lock the business process, and one to unlock the business process after the necessary operations have completed.</p>

Field	Description
DURATION	Time, in milliseconds, that the lock is applied. Required. The lock will time out or expire after this time. Note: If a business process halts due to an error, the service configuration will remain locked until you manually release the lock or restart the Application.
LOCK_KEY	The key for obtaining the lock. Required. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.
MAX_MEM_LOCK_PER_NODE_LISTED	Maximum number of local locks returned from each node. Valid value is any number greater than 0. Use this parameter if the value you specified in the ACTION field is List Locks. The default value is the value you specified in the defaultMaxMemLocksListedPerNode property in the centralops.properties.in file. Optional.
USER	User name associated with the lock (informational only). Optional. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.

Parameter(s) That Must be Added in BPML

The following additional parameter(s) can be used with Lock service by editing the BPML:

Parameter	Description
CLEAR_ON_START_UP	Clears the lock after you restart the Application. Use when you want to ensure that the lock is always clean when the Application is restarted. Optional. Valid values are true and false.
TimeStamp-MilliSeconds	Determines the time output in milliseconds since January 01, 1970, midnight.

Business Process Example

The following examples illustrate how the Lock service could be used in a business process.

Example Business Process 1

In this business process, a lock is applied by the Lock service. The lock is set to a duration of 600,000 milliseconds, and uses "Lock1" as the lock key. In this business process, the resource being locked is an instance of the Sleep service. Note that the sleep interval (duration) is 45 seconds. The Sleep service is followed by a second instance of the Lock service that releases the lock, thereby freeing up this configuration of the Sleep service configuration for other processes.

Note: The business process also includes onFault activities and messages in the event that one of the lock activities fails.

```
<process name="LockExample">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
    </operation>
    <output message="Xout">

```

```

    <assign to="DURATION">60000</assign>
    <assign to="LOCK_KEY">Lock1</assign>
    <assign to="CLEAR_ON_START_UP">true</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

<sequence name="Process">
  <operation name="Sleep">
    <participant name="TestSleepService"/>
    <output message="Xout">
      <assign to="SLEEP_INTERVAL">45</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <sequence name="UnLock">
    <operation name="UnLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="ACTION">unlock</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <onFault>
      <assign to="UnLock_Msg" append="true">Failed to obtain an unlock!</assign>
    </onFault>
  </sequence>
  <onFault>
    <operation>
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="ACTION">unlock</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </onFault>
</sequence>
<onFault>
  <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>

```

```

    </onFault>
  </sequence>
</process>

```

Example Business Process 2

In this business process, the time output is displayed in milliseconds since January 01, 1970, midnight.

```

<process name="LockExample">
  <rule name="Lock More than 2 hours old">
    <condition>number(/ProcessData/DateTime-MilliSeconds/text()) -
number(/ProcessData/debug_timestamp/text()) > 7200000</condition>
  </rule>

  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="LOCK_KEY" from="'LockTest'"/></assign>
        <assign to="ACTION">LOCK</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="USER" from="'TestUser'"/></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
        <assign to="Status_Report" from="Status_Rpt('Report')"/></assign>
      </input>
    </operation>

    <!-- Force the BP to fail -->
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="LOCK_KEY" from="'LockTest'"/></assign>
        <assign to="ACTION">LOCK</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="USER" from="'TestUser'"/></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
        <assign to="Status_Report" from="Status_Rpt('Report')"/></assign>
      </input>
    </operation>

    <onFault code="LOCK:Lock exists">
      <sequence name="Sequence Start">
        <operation name="ListLock">
          <participant name="LockService"/>
          <output message="Xout">
            <assign to="ACTION">LIST_LOCKS</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="Xin">

```

```

        <assign to="." from="DocToDOM(PrimaryDocument,'false')
/InMemoryLocks"></assign>
    </input>
</operation>

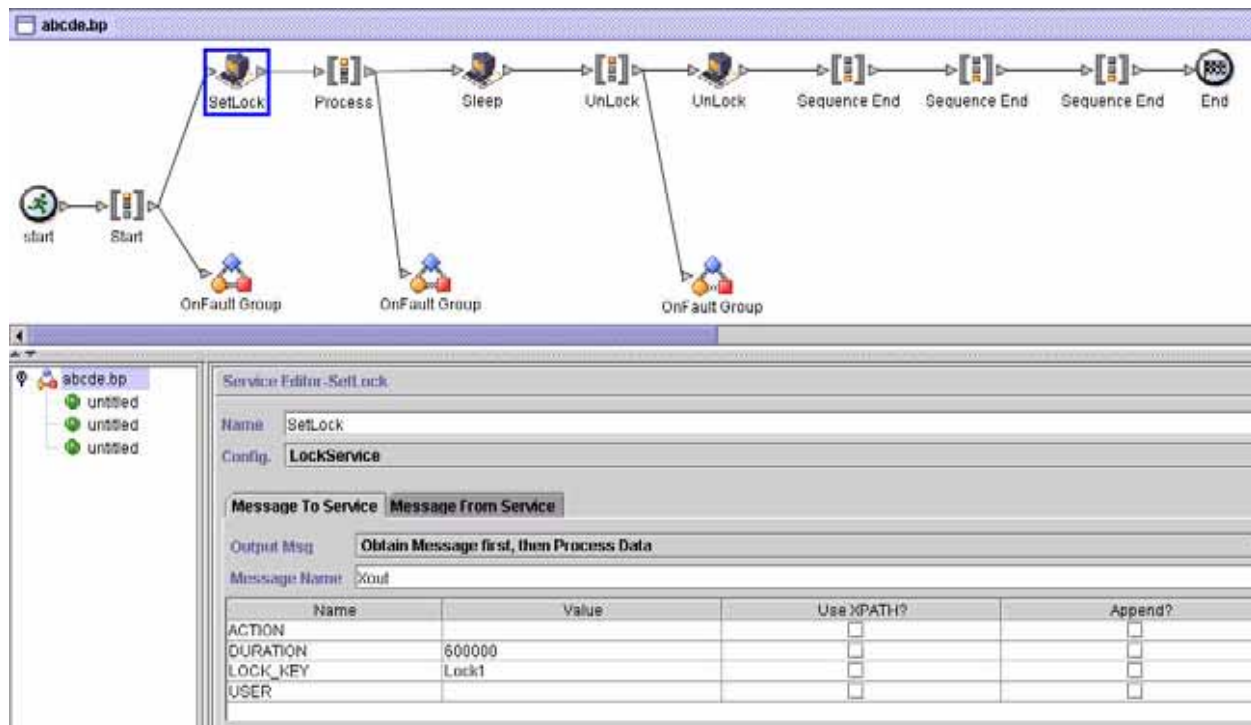
<operation name="Timestamp Utility">
    <participant name="TimestampUtilService"/>
    <output message="TimestampUtilServiceTypeInputMessage">
        <assign to="action">current_time</assign>
        <assign to="format">yyyyMMddHHmmssSSS</assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
        <assign to="/ProcessData/DateTime" from="/inmsg/time/text()"
append="true"></assign>
        <assign to="/ProcessData/DateTime-MilliSeconds"
from="/inmsg/currentTimeMillis/text()" append="true"></assign>
    </input>
</operation>

    <assign name="Assign" to="debug_timestamp"
from="/ProcessData/InMemoryLocks/NodeLocks/LockEntry[ResourceName/text() =
'LockTest']/TimeStamp/text()"></assign>
    <assign name="Assign" to="debug_timestamp_millise"
from="/ProcessData/InMemoryLocks/NodeLocks/LockEntry[ResourceName/text() =
'LockTest']/TimeStamp-MilliSeconds/text()"></assign>

    <choice name="Choice Start">
        <select>
            <case ref="Lock More than 2 hours old" activity="SendErrorMessage"/>
        </select>
        <sequence name="SendErrorMessage">
            <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
        </sequence>
    </choice>
</sequence>
</onFault>
</sequence>
</process>

```

The following GPM screen shows the example business process graphically. Note that the instance of the Lock service called SetLock is selected and its properties are displayed in the Service Editor in the lower half of the screen:



Lock Service – Frequently Asked Questions

How do I determine what the duration of a lock should be?

Set the lock duration value carefully (generally 2-3 times what the estimated time of execution will be) so that lock does not time out before the business process reaches the unlock step.

If the lock or unlock step fails (or any step in the business process between the two Lock services), are there any “cleanup” activities that need to be done?

Halted business processes can be terminated, or they can be left in halted state until the issue is resolved, and then restarted or resumed (as appropriate for the persistence level). The path to take depends on the needs of your business with regard to the business process itself. For example, is it mission-critical? Do other processes depend on its completion? Is the locked resource going to cause other business processes that use it to halt? Was the error caused by a problem in the configuration of the locked resource? Will this need to be corrected before using it again?

There are two general steps to follow first:

1. Check the lock manager page to see what’s locked.
2. Check **Business Process > BP Monitor > Current Processes** for more information about the error.

Then, once you have determined what caused the error, you can decide when to release the lock and when to terminate, restart, or resume the business process.

To manually release a locked resource:

Go to **Operations > Lock Manager**, and click **Go!** in the List panel. The locked resources are displayed on a results page. Locate the resources from your business process that are locked and clear the Lock checkbox for the resources.

To terminate, restart, or resume a business process:

Go to **Business Process > Monitor > Current Processes** and select the ID of the halted instance of your business process. From the page displayed, you can select the appropriate action for this business process: terminate, restart, or resume.

Are there any best practices for using the Lock service in a business process?

Use the Lock service twice in a business process—one to lock resources and one to unlock them. The first instance precedes the resources to be locked and the second instance follows them. See the *Business Process Example* on page 889 for a graphical representation. Do not use just one instance of the Lock service in a business process and let it expire instead of using a second Lock service to release the lock.

Mail Mime Service

The following table provides an overview of the Mail Mime service:

System name	MailMimeService
Graphical Process Modeler (GPM) categories	All Services, Communications
Description	<p>The Mail Mime service is used to construct a MIME message or to parse a MIME message.</p> <p>To construct a MIME message for outgoing e-mail, the service picks up the input data from the primary document. This becomes the body of the e-mail. Attachment data may be read from the file system, the document area, or both. All data is merged together into a single multipart MIME message. The resulting MIME message is placed in the primary document.</p> <p>The MIME message is then typically sent to the SMTP adapter to mail it.</p> <p>When parsing incoming e-mail, this service is typically preceded by the B2B Mail Client adapter, which actually receives the message, and passes it along to this service using the primary document. This service picks up the MIME message, including attachments, from the primary document. The primary message (body) is extracted and replaces the primary document. The rest of the messages (attachments) are placed in the document area.</p> <p>Each incoming attachment is referenced by a unique key. This key has the format "Mail_Mime_DOC_n", where "n" is the message count starting from 2 onwards. Message 1 is assumed to be the primary message. The service extracts the attachment MIME type and other attachment header information for use by other services.</p>
Business usage	This service can be used to construct a MIME message, or to parse a MIME message and extract the message content for further processing.
Usage example	Your company receives MIME messages from another business. You could use the Mail Mime service to parse these messages as part of a business process.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none">◆ SMTP Send adapter◆ B2B SMTP Client adapter◆ B2B Mail Client adapter
Application requirements	None
Initiates business processes?	No
Invocation	Invoked by an internal service

Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	Due to the use of a fully-qualified path name, this service is restricted to picking up files it can access directly and on specifically named systems.
Persistence level	System default
Testing considerations	<p>You can use the business process examples later in this section to test a configuration of the service.</p> <p>The problems encountered most frequently are:</p> <ul style="list-style-type: none"> ◆ File does not exist in the file system ◆ Service is not active

How the Mail Mime Service Works

The Mail Mime service is used for building or parsing a MIME message.

To build a MIME message:

1. The Mail Mime service picks up the input data from the primary document or from the file system or a combination of both.
2. The newly-built MIME message is returned as the primary document.

To parse a MIME message:

1. The Mail Mime service picks up the MIME message from the primary document and extracts the message contents to the primary document and to the document area of process data.
2. The content of the first message is returned as the primary document and the content of the subsequent messages is output to the document area of process data. These messages are each referenced by a unique key with the format Mail_Mime_DOC_*n*, where *n* is the message count, starting with the number 2.

Implementing the Mail Mime Service

To implement the Mail Mime service for use in a business process:

1. Create a Mail Mime service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mail Mime service. For information, see *Configuring the Mail Mime Service* on page 896.
3. Use the Mail Mime service in a business process.

Configuring the Mail Mime Service

When configuring the Mail Mime service, consider the following:

The parameter mail-mime-attachment(*n*) must include a fully qualified path to the file.

The file system provides all header and MIME information needed when using files as attachments. No configuration of attachment information (other than filename) is required within Application.

The fields in the following tables apply to building MIME messages, except mail-mime operation, which applies to parsing MIME messages. When the operation is parse, all values are read from the incoming message.

Application Configuration

The following table describes the fields used to configure the Mail Mime service in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Select the operation type (mail-mime-operation)	Specifies whether to build or parse a message. Optional. Valid values are: <ul style="list-style-type: none">◆ Construct (build) a new message◆ Parse a message The default is Construct (build) a new message. Note: If you are writing a business process without using the GPM, use the term “build” instead of “construct”.

GPM Configuration

The following table describes the fields used to configure the Mail Mime service in the GPM:

Field	Description
Config	Name of the service configuration.
mail-mime-bcc	The bcc field of the MIME message. Valid value is valid E-mail address. Optional for build.
mail-mime-body	Whether to get input data from the primary document. Valid values are True and False. Default is True. Optional for build.
mail-mime-cc	The cc field of the MIME message. Valid value is valid E-mail address. Optional for build.
mail-mime-recipient	Recipient of the MIME message. Valid value is valid E-mail address. Optional for build.

Field	Description
mail-mime-sender	Originator of the MIME message. Valid value is valid E-mail address. Optional for build.
mail-mime-subject	Subject field of the MIME message. Optional for build.
mail-mime-subject-encoding	Encoding type of the subject. Valid value is valid encoding type. Optional for build.
mail-mime-use-doc-area	Include documents in the document area of process data as attachments. All documents will be collected. Valid values are true and false. Default is false. Optional for build.
Parse	Not used. Leave at default.
Note: All of the previous parameters (with the exception of mail-mime-operation) apply to building MIME messages. If the operation type is parse, all values will be read from the incoming message.	
Note: The following parameters are not displayed in the GPM by default, however, you can manually add these parameters when necessary.	
mail-mime-attachment-count	Total number of attachments to be picked up from the file system. Optional.
mail-mime-attachment(n)	To specify each attachment, where <i>n</i> is the attachment count such as 1, 2, 3.... <i>n</i> . Valid value is a fully-qualified path to a file on the file system. Optional.

Parameters Passed from Business Process to Service

The following parameters can be passed to the service when started from a business process:

Note: If you need to overwrite a value that you have already configured in Application, you can pass the new value from the business process through the following parameters.

Parameter	Description
mail-mime-attachment-count	Total number of attachments to be picked up from the file system. Optional.
mail-mime-use-doc-area	Include documents in the document area of process data as attachments. All documents will be collected. Valid values are True and False. Default is False. Optional for build.
mail-mime-attachment(n)	To specify each attachment, where <i>n</i> is the attachment count. Valid value is a fully-qualified path to a file on the file system. Optional.
mail-mime-set-attachment	To specify if the contents of the primary document are set as an attachment in the MIME message. Valid values are True and False, and mail-mime-body must = True.
mail-mime-body	Whether to get input data from the primary document. Valid values are True and False. Default is True. Optional for build.
mail-mime-sender	Originator of the MIME message. Valid value is valid E-mail address. Optional.
mail-mime-recipient	Recipient of the MIME message. Valid value is valid E-mail address. Optional.

Parameter	Description
mail-mime-cc	The cc field of the MIME message. Valid value is valid E-mail address. Optional.
mail-mime-bcc	The bcc field of the MIME message. Valid value is valid E-mail address. Optional.
mail-mime-subject	Subject field of the MIME message. Optional.
mail-mime-subject-encoding	Encoding type of the subject. Valid value is valid encoding type. Optional.

How the Mail Mime Service Handles Attachment Data

For the build operation, the Mail Mime service can retrieve attachment data from the file system or the business process context. If the attachment data is in the file system, the service automatically looks up the content type and subtype of the data from the file system repository. If the attachment data is in the business process context, the filename and the content type/subtype information is read from the business process document. It is the responsibility of the calling business process to pass this information to the service. You can use the Get Document Info service to set or get the information.

The parse operation reads all information from the incoming e-mail and its attachments. Each attachment has data available for its name, MIME type and subtype, and character encoding. This information is made available when the attachments are written to the document area of process data.

Process Data Examples

When the Mail Mime service parses a MIME message, the total count of messages (including attachments) is output to the process data, in the format shown in the following example:

```
<ProcessData>
  <Mail_Mime>
    <Total_Message_Content>
      {---value----}
    </Total_Message_Content>
  </Mail_Mime>
</ProcessData>
```

The following example of process data is created when the Mail Mime service parses a MIME message with four message parts (main body and three attachments):

```
<ProcessData>
  <PrimaryDocument SCIObjectID="prodsys3:123c99:f88fee0b27:-796e" />
  <Mail_Mime_DOC_2 SCIObjectID=" prodsys3:123c99:f88fee0b27:-796c" />
  <Mail_Mime_DOC_3 SCIObjectID=" prodsys3:123c99:f88fee0b27:-796a" />
  <Mail_Mime_DOC_4 SCIObjectID=" prodsys3:123c99:f88fee0b27:-7968" />
  <Mail_Mime>
    <Total_Message_Content>4</Total_Message_Content>
  </Mail_Mime>
</ProcessData>
```

The first message content is put into the primary document and the attachments are placed in the document area, which can be referenced by a unique index key. The format of the unique index key is Mail_Mime_DOC_*n*, where *n* is the message count. The sample above shows that a total of four message parts is extracted from the MIME message. The first part is put in the primary document. The attachments are put into the document area, and they can each be referenced by a unique index key such as Mail_Mime_DOC_2, Mail_Mime_DOC_3 and Mail_Mime_DOC_4 respectively.

Business Process Examples

This section contains sample business processes that illustrate using the Mail Mime service to build or parse messages.

Building a Mime Message

The following business process illustrates building a MIME message from the primary document and sending the MIME message to an SMTP server, 00.000.00.00.

```
<process name="Test_mm_build">
<sequence name="optional">
<operation name="Build a mime message">
<participant name="MailMimeService"/>
<output message="Xout">
<assign to="." from="*"></assign>
<assign to="mail-mime-operation">build</assign>
<assign to="mail-mime-body">>true</assign>
<assign to="mail-mime-sender">ediintout@joeuser.mycompany.com</assign>
<assign to="mail-mime-recipient">steveuser@mycompany.com</assign>
<assign to="mail-mime-subject">This is my subject</assign>
</output>
<input message="Xin">
<assign to="." from="*"></assign>
</input>
</operation>
<operation name="One">
<participant name="SMTP_SEND_ADAPTER"/>
<output message="Xout">
<assign to="." from="*"></assign>
<assign to="xport-smtp-mailhost">00.000.00.00</assign>
<assign to="xport-smtp-mailport">25</assign>
<assign to="b2b-raw-message">>true</assign>
</output>
<input message="Xin">
<assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</process>
```

The following business process illustrates building a MIME message with 12 attachments. Each attachment is picked up from the file system:

```
<process name="Test_mm_build">
  <sequence name="optional">
```

```

    <operation name="One">
      <participant name="MailMimeService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="mail-mime-operation">build</assign>
        <assign to="mail-mime-body">>false</assign>
        <assign to="mail-mime-attachment-count">12</assign>
        <assign
to="mail-mime-attachment1">/home/testsys/test_data/smc1.props</assign>
        <assign
to="mail-mime-attachment2">/home/testsys/test_data/smc2.props</assign>
        <assign
to="mail-mime-attachment3">/home/testsys/test_data/smc3.props</assign>
        <assign
to="mail-mime-attachment4">/home/testsys/test_data/smc4.props</assign>
        <assign
to="mail-mime-attachment5">/home/testsys/test_data/smc5.props</assign>
        <assign
to="mail-mime-attachment6">/home/testsys/test_data/smc6.props</assign>
        <assign
to="mail-mime-attachment7">/home/testsys/test_data/smc7.props</assign>
        <assign
to="mail-mime-attachment8">/home/testsys/test_data/smc8.props</assign>
        <assign
to="mail-mime-attachment9">/home/testsys/test_data/smc9.props</assign>
        <assign
to="mail-mime-attachment10">/home/testsys/test_data/smc10.props</assign>
        <assign
to="mail-mime-attachment11">/home/testsys/test_data/smc11.props</assign>
        <assign
to="mail-mime-attachment12">/home/testsys/test_data/smc12.props</assign>
        <assign to="mail-mime-sender">ediintout@companyname.com</assign>
        <assign to="mail-mime-recipient">ediintin@companyname.com</assign>
        <assign to="mail-mime-subject">This is a mime message with 12
attachments</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Parsing a Mime Message

The following business process example illustrates parsing a MIME message. The Mail Mime service picks up the MIME message from the primary document, extracts all the message contents, and outputs them to the primary document and the document area:

```

<process name="Test_mm_parse">
  <sequence name="optional">
    <operation name="One">
      <participant name="MailMimeService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>

```

```
    <assign to="mail-mime-operation">parse</assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</process>
```

Mailbox Add Service

The following table provides an overview of the Mailbox Add service:

System name	Mailbox Add Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > Mailbox
Description	Adds a message to one or more mailboxes.
Business usage	Use this service to add a message to a mailbox.
Usage example	A trading partner sends a business document to a receiver requiring action.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms for Application
Related services	<p>The Mailbox Add service works with the other Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	None
Initiates business processes?	No
Invocation	A user with appropriate permissions to access the target mailbox must run the business process.
Business process context considerations	The user ID is extracted from the business process context at run time to confirm authorization of the business process to add messages to the target mailbox.

Returned status values	<p>Possible status values:</p> <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ User Permission Error – The user associated with the business process does not have appropriate permission to use the target mailbox. ◆ Invalid Mailbox Error – The target mailbox identified does not exist. ◆ Invalid Mailbox Parameter Error – An error occurred in passing parameters to this service. ◆ Multiple Extractable Types Specified By Business Process Error – The business process has improperly requested multiple extractability policies for the message. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred. ◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.
Restrictions	<p>Limitations for users and groups.</p> <p>When messages are added to a mailbox, message names are limited to 100 characters on DB2 and z/OS platforms. All other platforms support 255 character message names.</p>
Testing considerations	<p>Troubleshooting information for this service can be found in the Mailbox log files.</p>

How the Mailbox Add Service Works

Use the Mailbox Add service to add messages to a mailbox.

Business Process Example

The following BPML adds the current primary document to the mailbox named Acme with a message content type of ASCII. This business process will succeed only if the user associated with it has permissions for the Acme mailbox.

```
<!-- Add Msg to Acme mailbox -->
  <operation name="Mailbox Add Service">
    <participant name="MailboxAdd" />
    <output message="AddRequest">
      <assign to="." from="*"></assign>
      <assign to="MailboxPath">/Acme</assign>
      <assign to="ContentType">ascii</assign>
    </output>
    <input message="inmsg">
      <assign to="AddResults" from="*"></assign>
    </input>
  </operation>
```

Implementing the Mailbox Add Service

To implement the Mailbox Add service for use in a business process:

1. Create a Mailbox Add service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Add service. For information, see *Configuring the Mailbox Add Service* on page 905.

3. Use the Mailbox Add service in a business process.

Configuring the Mailbox Add Service

GPM Configuration

The following table describes the fields used to configure the Mailbox Add service in the GPM:

Consider the following when providing the MessageName and DocumentID:

If a primary document is available in the business process for the Mailbox Add service, DocumentID and MessageName are not required.

If no primary document is available in the business process for the Mailbox Add service, (OutputMsg is set to MessageOnly), DocumentID is required.

Field	Description
Config	Name of the adapter configuration.
ContentType	Indicates the MIME type and subtype. Use the following format: <i>MIME Type/MIME Subtype</i> .
DocumentId	Valid document ID for an existing document.
Extractable	Indicates whether this message can be extracted. Valid values are Yes and No. Note: You must provide a value for this field or one of the following: <ul style="list-style-type: none">◆ ExtractableCount◆ ExtractableUntil
ExtractableCount	Number of times this message may be accessed. Valid values are greater than or equal to 0. Note: You must provide a value for this field or one of the following: <ul style="list-style-type: none">◆ Extractable◆ ExtractableUntil
ExtractableUntil	Last date and time that this message may be extracted. Valid values are dates and times in <i>yyyyMMddThhmm</i> format. Note: You must provide a value for this field or one of the following: <ul style="list-style-type: none">◆ ExtractableCount◆ Extractable
MailboxPath	Mailbox path that the user has permission to use. If a user has been set up with a virtual root, the MailboxPath will be a relative path to the virtual root. The virtual root is not visible to the business process. Required. Valid values are UNIX-style paths where the folders correspond to a mailbox name. A path must begin with the '/' character and use the '/' to separate mailboxes in the hierarchy. Following is an example of a valid mailbox path: /Customers/DallasHardware/Inbound The following characters cannot be used in a mailbox name: ? < > : \ / % *.

Field	Description
MessageName	User-defined. The following characters cannot be used in a message name: \ / : * ? " < > % ! If you do not provide a value for MessageName, Application uses the document name of the document referred to in the DocumentId parameter passed to the service.

Output from Service to Business Process

The following table contains the parameters passed from the Mailbox Add service to a business process:

Parameter	Description
Input Msg	Identifies how to gather information from the business process. <ul style="list-style-type: none"> ◆ Allow process data write – The message response from the service will be written to the process data. ◆ Allow message write – Allows the service to write to the message.
MessageId	ID of the stored message.
DocumentId	Document ID corresponding to the stored message.
CreateDateTime	Creation date.
MessageName	User-defined. The following characters cannot be used in a message name: \ / : * ? " < > % ! If you do not provide a value for MessageName, Application uses the document name of the document referred to in the DocumentId parameter passed to the service.
MailboxPath	Path of the mailbox which the message was added to. If the user has been set up with a virtual root, the MailboxPath will be a relative path to the virtual root. The virtual root is not visible to the business process.
ContentType	Indicates the MIME type and subtype. Use the following format: <i>MIME Type/MIME Subtype</i> .
MessageSize	Size of the added message in bytes.
ExtractableCount	Number of times this message may be accessed. One of the three extractability parameters is returned.
Extractable	Indicates whether this message can be extracted. One of the three extractability parameters is returned.
ExtractableUntil	The last date and time that this message may be extracted. One of the three extractability parameters is returned.

Mailbox Correlate Document Service

The following table provides an overview of the Mailbox Correlate Document service:

System name	Mailbox Correlate Document Service
Graphical Process Modeler (GPM) category	None
Description	Correlates the specified document to three separate components of the specified message: the message name, the message ID and the path of the mailbox. The correlation names for each of these components are Mailbox_MessageName, Mailbox_MessageId, and Mailbox_MailboxPath. The correlation for mailbox path will contain the absolute path of the mailbox the message was added to.
Business usage	This is a system service.
Usage example	The business process MailboxAS2Add correlates a packaged (enveloped) AS2 document with the message and mailbox that the corresponding payload is destined to. This allows the EDIINT detail UI page to also display the message and mailbox that an AS2 transaction corresponds to.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox Correlate Document service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	None

Invocation	Does not perform any permissions checking.
Business process context considerations	No information from the business process except what is supplied through the normal BPML <operation> and <assign> syntax.
Returned status values	<p>Possible status values:</p> <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ Invalid Mailbox Parameter Error – An error occurred passing parameters to this service such as a message ID in an invalid format. ◆ Message Not Found Error – A message does not exist that corresponds to the supplied message ID. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred. ◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.
Restrictions	The only restriction is that the supplied message ID and document ID must both be valid (they must correspond to an actual message and an actual document in Application).
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Correlate Document Service Works

Application Mailbox uses the Mailbox Correlate Document service to correlate the specified document to three separate component of the message: message name, message ID, and mailbox path.

Mailbox Delete Mailbox Service

The Mailbox Delete Mailbox service deletes one or multiple mailboxes, as well as submailboxes, messages, virtual roots, routing rules, and permissions. It is designed to completely and permanently remove mailboxes and everything associated with them.

Note: There is a separate but similarly named service called the Mailbox Delete service that deletes only messages in mailboxes.

The following table provides an overview of the Mailbox Delete Mailbox service:

System Name	Mailbox Delete Mailbox Service
Graphical Process Modeler (GPM) categories)	Internet B2B, Mailbox
Description	This service deletes one or multiple mailboxes.
Business usage	A business user uses this service to delete mailboxes that meet specific criteria.
Usage example	Dolnet currently has 100 trading partners defined in the Application system that exchange data with the company. Each trading partner has its own mailbox and unique way of transferring data. Dolnet has decided to stop any further transactions with one of the trading partners, Vialore, and must now modify Application such that Vialore can no longer exchange data with Dolnet. As part of this change, Vialore's mailbox will have to be deleted. The Mailbox Delete Mailbox service is used in a business process to delete Vialore's mailbox.
Preconfigured?	No
Requires third party files?	No third-party files are required.
Platform availability	All supported Application platforms
Related services	<p>This service also works with the other Mailbox services to provide a mailbox capability to Application. The related services are:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Enables a business process to notify the mailbox system that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to notify the Mailbox service of a failed message extraction.◆ Mailbox Query service – Queries messages.◆ Mailbox Delete service – Deletes messages.◆ Mailbox Update service – Updates messages.◆ Mailbox Evaluate All Automatic Rules service – Schedules automatic routing rules for the Mailbox system.◆ Mailbox Evaluate Routing – Applies mailbox routing rules from a business process.

Application requirements	Nothing external to Application is required for the use of this service.
Initiates business processes?	No
Invocation	A user with the Mailbox Global Delete permission (obtained through the addition to a User Account or Group through the Accounts page) must execute the business process invoking this service.
Business process context considerations	The Mailbox Global Delete permission ensures that the business process is authorized to access the referenced mailboxes.
Returned status values	<p>Possible status values that can be returned from this service are:</p> <ul style="list-style-type: none"> ◆ Success – Normal execution with the return parameters specified below. ◆ User Permission Error – The user associated with the business process does not have Mailbox Global Delete permission. ◆ Invalid Mailbox Error – The mailbox identified as the delete target does not exist. ◆ Invalid Mailbox Parameter Error – An error has been made in the passing of parameters to use this service, possibly a malformed date or numeric value. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository has occurred. ◆ Mailbox Service Error – A generic error associated with the Mailbox service has occurred.
Restrictions	The limitations of this service are based on the assignment of the Mailbox Global Delete permission.
Testing considerations	Debug information for this service can be found in the mailbox log files.

How the Mailbox Delete Mailbox Service Works

1. Starting with the originally passed list of mailboxes to delete, the Mailbox Delete Mailbox service recursively adds each underlying submailbox until every mailbox in the hierarchy is represented.
2. Then for each mailbox, the service deletes all contained messages, the permission, all of the permission associations to users or groups, virtual roots, routing rules, and the mailbox itself.
3. If there are multiple mailboxes in the routing rule, the deleted mailbox is removed from the rule, otherwise the rule will be deleted.
4. All operations are performed together as a group, all or nothing. If any part fails, no changes are made by the service and an error is returned.

Implementing the Mailbox Delete Mailbox Service

To implement the Mailbox Delete Mailbox service, complete the following tasks:

1. Obtain Mailbox Global Delete permission, through the addition to a User Account or Group in the Accounts page.

2. Create a configuration of the Mailbox Delete Mailbox service. For basic information about creating service configurations, see *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Mailbox Delete Mailbox Service* on page 911.
3. Add the provided Mailbox Delete Mailbox service configuration to a business process.
4. Test and run the business process.

Note: A preconfigured Mailbox Delete Mailbox service is also provided.

Configuring the Mailbox Delete Mailbox Service

You must specify field settings in the Admin Console and the Graphical Process Modeler.

Creating a Service Configuration in the Admin Console

Use the field definitions in the following table to set up the configuration provided with Application.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about service groups, see <i>Managing Services and Adapters</i>.</p>

Defining the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Parameter	Description
Force	<p>Whether a mailbox delete should be forced when the mailbox contains submailboxes or messages. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ Yes – Mailbox is deleted even if it has messages or submailboxes. ◆ No – Mailbox is not deleted if it has messages or submailboxes. <p>Note: If no value is specified, the service will default to use No.</p>

Parameter	Description
MailboxPath	<p>Mailbox path that you would like to delete. If you have been set up with a virtual root, then MailboxPath will be a relative path to the virtual root. The virtual root is not visible to the business process.</p> <p>Valid value is a Unix path where the folders correspond to a mailbox hierarchy. The path must begin with the / character and use the / to delimit mailboxes in the hierarchy. The space character is permitted in the middle of a mailbox name, and the following characters are not permitted in a mailbox name: ?, <, >, , ", :, \, /, %, and *.</p> <p>Note: This element can be repeated to delete multiple mailboxes.</p>
Userld	<p>Do not use.</p> <p>Note: The Userld parameter has no effect on this service and is unsupported. The service always uses the user ID that is running the business process.</p>

Business Process Example

The following example shows how the Mailbox Delete Mailbox service can be used in a business process.

Business Scenario

Dolnet currently has 100 trading partners defined in the Application system that exchange data with the company. Each trading partner has its own mailbox and unique way of transferring data. Dolnet has decided to stop any further transactions with one of the trading partners, Vialore, and must now modify Application such that Vialore can no longer exchange data with Dolnet. As part of this change, Vialore's mailbox will have to be deleted. The Mailbox Delete Mailbox service is used in a business process to delete Vialore's mailbox.

This is the BPML for the business process:

```
<process name="MailboxDeleteMailboxProcess">
  <sequence name="MailboxDeleteMailboxSequence">
    <operation name="Mailbox Delete Mailbox Service">
      <participant name="MailboxDeleteMailbox"/>
      <output message="DeleteRequest">
        <assign to="MailboxPath">/VialoreMailbox</assign>
        <assign to="Force">Yes</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="DeleteResponse">
        <assign to="DeleteResponse" from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```


Parameters Passed From Service to Business Process

The following table contains the parameters passed from the Mailbox Delete Mailbox service to the business process:

Parameter	Description
DeleteResponse	Specifies all resources deleted as a result of operation of the service, including mailboxes, submailboxes, messages, permissions, virtual roots, and routing rules.

Following is an example of DeleteResponse in the process data:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <DeleteResponse>
    <Mailbox>
      <MailboxId>11</MailboxId>
      <MailboxPath>/MyMailbox</MailboxPath>
    </Mailbox>
    <Mailbox>
      <MailboxId>12</MailboxId>
      <MailboxPath>/MyMailbox/MyMailboxSub</MailboxPath>
    </Mailbox>
    <Message>
      <MessageId>9</MessageId>
      <MessageName>delta.txt</MessageName>
    </Message>
    <Message>
      <MessageId>10</MessageId>
      <MessageName>delta2.txt</MessageName>
    </Message>
    <Permission>
      <PermissionName>/MyMailbox.mbx</PermissionName>
    </Permission>
    <Permission>
      <PermissionName>/MyMailbox/MyMailboxSub.mbx</PermissionName>
    </Permission>
    <Rule>
      <RuleId>4f06da5c:1057842a730:2327</RuleId>
    </Rule>
    <VirtualRoot>
      <UserId>jane_doe</UserId>
    </VirtualRoot>
  </DeleteResponse>
</ProcessData>
```

Mailbox Delete Service

The following table provides an overview of the Mailbox Delete service:

System name	Mailbox Delete Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > Mailbox
Description	Deletes messages in one or many mailboxes.
Business usage	Use this service to delete mailbox messages that meet specific criteria. Call this service in a business process to delete messages that are no longer needed.
Usage example	A business process uses the Mailbox Delete service to delete a message it has just processed and is unnecessary.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported application platforms
Related services	<p>The Mailbox Delete service works with the other mailbox services in the application to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal an application mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to an application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for an application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to the application is required to use this service.
Initiates business processes?	No
Invocation	The business process using this service must be run by a user with permission to access all applicable mailboxes. A user with Mailbox Global Delete permission can delete any message from any mailbox.
Business process context considerations	The service extracts the user permissions at run time to confirm authorization of the business process to access the referenced mailboxes.

Returned status values	<p>Possible status values:</p> <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ User Permission Error – The user associated with the business process either does not have Mailbox Global Delete permission or does not have permission on the target mailboxes. ◆ Invalid Mailbox Error – Target mailbox does not exist. ◆ Invalid Mailbox Parameter Error – An error occurred passing parameters to this service, such as a message ID in an invalid format. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred. ◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.
Restrictions	The limitations of this service are based on the assignment of mailbox permissions to users and groups. The Mailbox Delete service can be used by users assigned the Mailbox Global Delete permission.
Testing considerations	Troubleshooting information for this service can be found in application Mailbox log files.

How the Mailbox Delete Service Works

Use the Mailbox Delete service to delete messages from one, many, or all mailboxes. The user ID that the business process is using must have access to the mailbox you are deleting from.

Business Process Example

The following BPML deletes all messages currently in the Acme mailbox. If the user running this business process has been assigned a virtual mailbox root, then the Acme mailbox path will be relative to the users virtual mailbox root.

The user must have permission for the Acme mailbox, or has the Mailbox Global Delete permission.

```
<process name="ACMEDelete">
  <sequence name="Delete">
    <!--Delete Messages in the Mailbox -->
    <operation name="Mailbox Delete Service">
      <participant name="MailboxDelete"/>
      <output message="DeleteRequest">
        <assign to="." from="*"></assign>
        <assign to="MailboxPath">/Acme</assign>
      </output>
      <input message="inmsg">
        <assign to="DeleteResults" from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Implementing the Mailbox Delete Service

To implement the Mailbox Delete service for use in a business process:

1. Create a Mailbox Delete service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Delete service. For information, see *Configuring the Mailbox Delete Service*.
3. Use the Mailbox Delete service in a business process.

Configuring the Mailbox Delete Service

The following table describes the fields used to configure the Mailbox Delete service in the application Admin Console and in the GPM.

Field	Description
DaysOld	Number of days a message can exist in the application Mailbox before being deleted. Required. Number must be >0.
EndDateTime	Defines the latest allowable value for this query. Valid values are dates and times in <i>yyyyMMddThhmm</i> format.
MailboxPath	Path of the mailbox where the message resides. If the user has been set up with a virtual root, the MailboxPath will be a relative path to the virtual root. The virtual root is not visible to the business process. Required.
MailboxSelection	Mailbox paths to be queried during message deletion. Required. Valid values are: <ul style="list-style-type: none">◆ All Mailboxes – Queries all mailboxes to determine messages to be deleted.◆ Selected Mailboxes – Displays a Select Mailboxes page that allows you specify which mailboxes to query. Note: Leaving this field blank will cause errors to be written to the status report.
MessageExtractable	Designates which messages to delete. Required. <ul style="list-style-type: none">◆ Delete Only unextractable messages◆ Delete All messages
MessageId	ID of the stored message.
MessageNamePattern	Pattern that determines message selection. If a message name matches the pattern, the corresponding message is selected. Required. Valid values: Alphanumeric character(s) or asterisk (*) used as a wildcard. The following characters cannot be used in a message name: ? < > / " % * Required format if a wildcard is used: x.y where x and y may be alphanumeric or a wildcard. Example: *.po Note: You cannot specify a message name pattern composed solely of a single asterisk. it will fail the adapter.
StartDateTime	Defines the earliest allowable value for this query. Valid values are dates and times in <i>yyyyMMddThhmm</i> format.
UserId	User ID associated with the business process. Note: The user ID must have access to the mailbox being queried.

Output from Service to Business Process

The following table contains the parameter passed from the Mailbox Delete service to a business process:

Parameter	Description
MessageId	ID of the stored message.

Mailbox Evaluate All Automatic Routing Rules Service

The following table provides an overview of the Mailbox Evaluate All Automatic Routing Rules service:

System name	Mailbox Evaluate All Automatic Routing Rules Service
Graphical Process Modeler (GPM) category	None
Description	Evaluates all automatic routing rules. Automatic routing rules are evaluated against all messages added to Application Mailbox since the previous evaluation. Messages are eligible for automatic routing rule evaluation only once. If a message is added to a mailbox and is not targeted by an automatic routing rule, that message will never be eligible again for automatic routing rule processing, even if an automatic routing rule is later created that would target the message. Messages added successfully to a mailbox are processed by the automatic routing rules even if the Application system fails and requires restarting.
Business usage	Create Mailbox routing rules and designate some of them to be automatic. All automatic routing rules are periodically evaluated by this service.
Usage example	The Application Mailbox system administrator creates a new automatic routing rule. The preconfigured version of the Mailbox Evaluate All Automatic Routing Rules service is enabled. The rule is automatically evaluated every minute. If a message arrives in the specified mailbox, the rule is triggered.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox Evaluate All Automatic Routing Rules service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No

Invocation	Not applicable (Internal service)
Business process context considerations	None
Returned status values	Possible status values: <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ Error – A generic error occurred.
Restrictions	This service has no restrictions.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Evaluate All Automatic Routing Rules Service Works

Application Mailbox uses the Mailbox Evaluate All Automatic Routing Rules service to automatically evaluate all automatic routing rules for messages that have not been routed. This service is disabled by default. You must enable it before it can be used.

Implementing the Mailbox Evaluate All Automatic Routing Rules Service

To implement the Mailbox Evaluate All Automatic Routing Rules service for use in a business process:

1. Create an Mailbox Evaluate All Automatic Routing Rules service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Evaluate All Automatic Routing Rules service. For information, see *Configuring the Mailbox Evaluate All Automatic Routing Rules Service* on page 920.
3. Use the Mailbox Evaluate All Automatic Routing Rules service in a business process.

Configuring the Mailbox Evaluate All Automatic Routing Rules Service

To configure the Mailbox Evaluate All Automatic Routing Rules service, you must specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Mailbox Routing Rule (RuleId)	A valid ID for a routing rule. Required.
Run as User	Enter (or select from the list) the user ID to be associated with business process instances of this service.
Use 24 Hour Clock Display	If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.
Schedule	<p>Information about scheduling the service configuration to run.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, this service does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the month Valid values are the day of the month (including the last day of the month (LDM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup.

Mailbox Evaluate Routing Rule Service

The following table provides an overview of the Mailbox Evaluate Routing Rule service.

System name	Mailbox Evaluate Routing Rule Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > Mailbox
Description	Evaluates a mailbox routing rule.
Business usage	Create a specific routing rule then use the Mailbox Evaluate Routing Rule service to schedule evaluation of that rule. This allows business processes to use routing rules. This service can be scheduled.
Usage example	The Application Mailbox system administrator creates several automatic Mailbox Routing Rules that target specific mailboxes to execute specific business processes. This service evaluates those routing rules on a schedule established by the system administrator using the Mailbox Evaluate Routing Rule service.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox Evaluate Routing Rule service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	This service can be scheduled. It also can be started by business processes wanting to trigger a Mailbox routing rule.
Business process context considerations	No

Returned status values	Possible status values: <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ Invalid Rule ID Error – The target ruleid could not be evaluated. ◆ Mailbox Router Error – A generic error associated with the mailbox router occurred. ◆ Mailbox service Error – A generic error associated with the mailbox service occurred.
Restrictions	The limitations of this service are based on the assignment of mailbox permissions to users and groups.
Testing Considerations	Troubleshooting for this services can be found in the Application Mailbox log files.

How the Mailbox Evaluate Routing Rules Service Works

Use the Mailbox Evaluate Routing Rule service to evaluate a routing rule that is defined in Application Mailbox. The evaluation can be scheduled.

Implementing the Mailbox Evaluate Routing Rule Service

To implement the Mailbox Evaluate Routing Rule service for use in a business process:

1. Create an Mailbox Evaluate Routing Rule service configuration.
2. Configure the Mailbox Evaluate Routing Rule service.
3. Use the Mailbox Evaluate Routing Rule service in a business process.

Configuring the Mailbox Evaluate Routing Rule Service

To configure the Mailbox Evaluate Routing Rule service, specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Mailbox Routing Rule (RuleId)	A valid ID for a routing rule. Required if RuleDescription is not provided.

Field	Description
RuleDescription	A description that uniquely identifies the routing rule for this service. Unlike RuleId, RuleDescription does not change as the result of an import. If set, RuleDescription takes precedence over RuleId and RuleId is ignored. Required if RuleId is not provided.
Run as User	Enter (or select from the list) the user ID to be associated with business process instances of this service.
Use 24 Hour Clock Display	If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.
Schedule	<p>Information about scheduling the service configuration to run.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, this service does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the month Valid values are the day of the month (including the last day of the month (LDOM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup.

GPM Configuration

The following table describes the fields used to configure the Mailbox Evaluate Routing Rule service in the GPM:

Field	Description
Config	Name of the service configuration.
RuleId	A valid ID for a routing rule. Required if RuleDescription is not provided.

Parameters Passed from Business Process to Service

Parameter	Description
RuleId	ID of the routing rule that was evaluated.

Parameter	Description
RuleDescription	A description that uniquely identifies the routing rule for this service. Unlike RuleId, RuleDescription does not change as the result of an import. If set, RuleDescription takes precedence over RuleId and RuleId is ignored. Required if RuleId is not provided.

Output from Service to Business Process

The following table contains the parameter passed from the Mailbox Evaluate Routing Rule service to a business process:

Parameter	Description
RuleId	ID of the routing rule that was evaluated.
MessageId	An ID of each of the Messages that were evaluated against rule with RuleId.

Business Process Example

The following BPML evaluates the rule corresponding to ruleId = 268ef9:f6af57c58f:-72bf:

```
<process name="LaunchAcmeRoutingRule">
  <sequence name="Evaluate">
    <!--Evaluate the AcmeRoutingRule -->
      <operation name="Mailbox Evaluate Routing Rule Service">
        <participant name="MailboxEvaluateRoutingRule"/>
        <output message="EvaluateRuleRequest">
          <assign to="." from="*"></assign>
          <assign to="RuleId">268ef9:f6af57c58f:-72bf </assign>
        </output>
        <input message="inmsg">
          <assign to="EvaluateRuleResults" from="*"></assign>
        </input>
      </operation>
    </sequence>
  </process>
```

Mailbox Extract Abort Service

The following table provides an overview of the Mailbox Extract Abort service:

System name	Mailbox Extract Abort Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > Mailbox
Description	Signifies to the business process the unsuccessful extraction of a message from a mailbox.
Business usage	Use this service to signify to the business process that the extraction of a mailbox message was not completed successfully, so the message information (the extractability count) should not change.
Usage example	If SAP processing fails after purchase orders are extracted from the mailbox and the Mailbox Extract Abort service is run, the message extractability count does not change.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox Extract Abort service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	The business process using this service must have performed a Mailbox Extract Begin on the target message.
Business process context considerations	The business process ID and branch ID must match the business process context that performed the Mailbox Extract Begin.

Returned status values	<p>Possible status values:</p> <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ User Permission Error – The user associated with the business process does not have permission to use the target mailbox. ◆ Invalid Mailbox Error – This might indicate changes in the assignment of virtual mailbox roots which would disallow the extract process. ◆ Invalid Mailbox Parameter Error – An error occurred passing parameters to this service such as a message ID in an invalid format. ◆ Message Not Found Error – The requested message cannot be found for extraction. ◆ Message Not Extractable Error – The target message's extraction policy forbids extraction. ◆ Mailbox Extract Invalid State Error – The current business process has not performed an Extract Begin. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred. ◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.
Restrictions	This service cannot be used against a target message until the Mailbox Extract Begin service is called for that message. Additionally, the business process that calls this service must have the same business process ID and branch ID as the business process that called the Mailbox Extract Begin service.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Extract Abort Service Works

Use the Mailbox Extract Abort service to recover from Mailbox Extract Begin service failures. The Mailbox Extract Abort service releases the lock that the business process has on the message. If the extract policy of the message is a count, this service returns the count to the value it was before the Mailbox Extract Begin service was run.

For example, the Mailbox Extract Begin service attempts to extract a message with an extractable count of one. The message is extractable and available, so the Mailbox Extract Begin service extracts the message and decrements the extractable count to zero. During the extraction, the communication protocol has an error. The Mailbox Extract Abort service is started. The Mailbox Extract Abort service increments the Extractable Count by one, allowing a subsequent business process to extract the message.

It resets the extractable count of the message to one so that the business process can attempt to extract the file the next time the Mailbox Extract Begin service is run.

Business Process Example

The following BPML aborts the extraction of message 1234 from Application Mailbox. If it is assumed message 1234 has been successfully extracted by business process ID=789, branch ID=456, the BPML will succeed if run by a business process with ID=789 and branch ID=456.

```
<operation name="Mailbox ExtractBegin Service">
  <participant name="MailboxExtractBegin"/>
  <output message="ExtractBeginRequest">
```

```

        <assign to="." from="*"></assign>
        <assign to="MessageId">1234</assign>
    </output>
    <input message="inmsg">
        <assign to="." from="*"></assign>
    </input>
</operation>
<!-- Abort Extract Operation. -->
<operation name="Mailbox ExtractAbort Service">
    <participant name="MailboxExtractAbort" />
    <output message="ExtractAbortRequest">
        <assign to="." from="*"></assign>
        <assign to="MessageId">1234</assign>
    </output>
    <input message="inmsg">
        <assign to="." from="*"></assign>
    </input>
</operation>

```

Implementing the Mailbox Extract Abort Service

To implement the Mailbox Extract Abort service for use in a business process:

1. Create an Mailbox Extract Abort service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Extract Abort service. For information, see *Configuring the Mailbox Extract Abort Service* on page 928.
3. Use the Mailbox Extract Abort service in a business process.

Configuring the Mailbox Extract Abort Service

GPM Configuration

The following table describes the fields used to configure the Mailbox Extract Abort service in the GPM:

Field	Description
Config	Name of the adapter configuration.
MessageId	ID of the stored message. Required. Valid values are numbers >=0.

Output from Service to Business Process

The following table contains the parameter passed from the Mailbox Extract Abort service to a business process:

Parameter	Description
MessageId	ID of the stored message.

Mailbox Extract Begin Service

The following table provides an overview of the Mailbox Extract Begin service:

System name	Mailbox Extract Begin Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > Mailbox
Description	Begins the extraction of a message from a mailbox.
Business usage	Use this service to begin extracting a message from a mailbox.
Usage example	Extract all purchase orders from a mailbox and deliver them to the SAP adapter for back-office processing.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox Extract Begin service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	User with appropriate permissions to access the target mailbox must run the business process.
Business process context considerations	The user permissions are extracted from the business process context at run time to confirm authorization of the business process to add messages to the target mailbox.

Returned status values	<p>The possible status values a service can return are:</p> <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ User Permission Error – The user associated with the business process does not have permission to use the target mailbox. ◆ Invalid Mailbox Error – The mailbox does not exist. ◆ Invalid Mailbox Parameter Error – An error occurred passing parameters to this service such as a message ID in an invalid format. ◆ Message Not Found Error – The requested message cannot be found for extraction. ◆ Message Not Extractable Error – The target message's extraction policy forbids extraction. ◆ Mailbox Extract Invalid State Error – The current business process has not performed an extract begin request. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred. ◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.
Restrictions	The limitations of this service are based on the assignment of mailbox permissions to users and groups.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Extract Begin Service Works

Use the Mailbox Extract Begin service to extract messages from a mailbox. This service checks the extractability and availability of a message and verify whether the user that is running the business process has permissions on the mailbox.

If the extractability is Extractable Count = 1, the Mailbox Extract Begin service locks the message until the extraction is complete. After the extraction is complete, the count is decremented. If the ExtractableCount = x , x number of Mailbox Extract Begin services can lock the message. You can use this service in one of the following modes:

Mailbox Extract Begin with CommitNow = No. This is the default mode. In this mode, you must include the Mailbox Extract Abort service and Mailbox Extract Commit service in your business process. This mode is useful to control the extract count. If an error occurs during the extraction, such as a protocol failure, the message count will reincrement back to the original count. The business process locks the message until the Mailbox Extract Commit service or the Mailbox Extract Abort service completes.

Mailbox Extract Begin with CommitNow = Yes. In this mode, all message extracts are extracted and committed in a single step. This mode is useful if you do not need control over the extract count. If an error occurs during the extract, such as a protocol failure, the message count will not be incremented back to the original count. The business process does not lock the message.

Business Process Example

The following BPML extracts the message 1234 from a Application mailbox. This business process will succeed if the user associated with this BPML has permission to the mailbox contains message 1234.

```
<!-- Read in the payload. -->
```

```

<operation name="Mailbox ExtractBegin Service">
  <participant name="MailboxExtractBegin"/>
  <output message="ExtractBeginRequest">
    <assign to="." from="*"></assign>
    <assign to="MessageId">1234</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

```

Implementing the Mailbox Extract Begin Service

To implement the Mailbox Extract Begin service for use in a business process:

1. Create an Mailbox Extract Begin service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Extract Begin service. For information, see *Configuring the Mailbox Extract Begin Service* on page 932.
3. Use the Mailbox Extract Begin service in a business process.

Configuring the Mailbox Extract Begin Service

The following table describes the fields used to configure the Mailbox Extract Begin service in the GPM:

Field	Description
Config	Name of the adapter configuration.
CommitNow	Yes – Automatically commits the message for extraction without the use of Mailbox Extract Commit service. The business process does not lock the message. No – Requires the Mailbox Extract Commit service to commit the message for extraction. The business process locks the message until the Mailbox Extract Commit or Mailbox Extract Abort service is complete. Default.
MessageId	ID of the stored message. Required. Valid values are numbers >=0.

Output from Service to Business Process

The following table contains the parameters passed from the Mailbox Extract Begin service to a business process:

Parameter	Description
MessageId	ID of the stored message.
DocumentId	Document ID corresponding to the stored message.
CreateDateTime	Creation date.

Parameter	Description
MessageName	Name of the message.
MailboxPath	Path of the mailbox which the message was added to. If the user has been set up with a virtual root, the MailboxPath will be a relative path to the virtual root. The virtual root is not visible to the business process.
ContentType	Indicates the MIME type and subtype. Use the following format: <i>MIME Type/MIME Subtype</i> .
MessageSize	Size of the added message in bytes.
ExtractableCount	Number of times this message may be accessed. One of the three extractability parameters is returned.
Extractable	Indicates whether this message can be extracted. One of the three extractability parameters is returned.
ExtractableUntil	The last date and time that this message may be extracted. One of the three extractability parameters is returned.

Mailbox Extract Commit Service

The following table provides an overview of the Mailbox Extract Commit service:

System name	Mailbox Extract Commit Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > Mailbox
Description	Indicates the successful completion of the extraction of a message from a mailbox.
Business usage	A trading partner would use this service to indicate the successful completion of a unit of work.
Usage example	After all purchase orders are extracted from the mailbox and successfully delivered to the SAP adapter for back-office processing, the message extractability count is permanently decremented.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox Extract Commit service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	The business process using this service must have already performed a Mailbox Extract Begin on the target message.
Business process context considerations	The business process ID and branch ID must match the business process ID and branch ID that performed the Mailbox Extract Begin.

Returned status values	<p>The possible status values a service can return are:</p> <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ User Permission Error – The user associated with the business process does not have permission to use the target mailbox. ◆ Invalid Mailbox Error – Indicates changes in the assignment of virtual mailbox roots, which would disallow the extract process. ◆ Invalid Mailbox Parameter Error – An error occurred passing parameters to this service such as a message ID in an invalid format. ◆ Message Not Found Error – The requested message cannot be found for extraction. ◆ Message Not Extractable Error – The target message's extraction policy forbids extraction. ◆ Mailbox Extract Invalid State Error – The current business process has not performed an Extract Begin. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred. ◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.
Restrictions	The Mailbox Extract Begin service must be called prior to this service. Additionally, the calling business process instance must be the same instance that called the Mailbox Extract Begin service.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Extract Commit Service Works

Use the Mailbox Extract Commit service to complete the extraction of a message from a mailbox that was started by the Mailbox Extract Begin service. This service releases the message that was locked by the Mailbox Extract Begin service. To use this service in your business process, you must include all three services: the Mailbox Extract Begin service, Extract Abort service, and Extract Commit service.

Business Process Example

The following BPML will commit the extraction of message 1234 from Application Mailbox. If it is assumed message 1234 has been successfully extracted by business process ID=789, branch ID=456, the BPML will succeed if run by a business process with ID=789 and branch ID=456.

```
<operation name="Mailbox ExtractBegin Service">
  <participant name="MailboxExtractBegin"/>
  <output message="ExtractBeginRequest">
    <assign to="." from="*"></assign>
    <assign to="MessageId">1234</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<!-- Commit Extract Operation. -->
<operation name="Mailbox ExtractCommit Service">
  <participant name="MailboxExtractCommit"/>
```

```

    <output message="ExtractCommitRequest">
      <assign to="." from="*"></assign>
      <assign to="MessageId">1234</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

```

Implementing the Mailbox Extract Commit Service

To implement the Mailbox Extract Commit service for use in a business process:

1. Create an Mailbox Extract Commit service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Extract Commit service. For information, see *Configuring the Mailbox Extract Commit Service* on page 936.
3. Use the Mailbox Extract Commit service in a business process.

Configuring the Mailbox Extract Commit Service

To configure the Mailbox Extract Commit service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the adapter configuration.
MessageId	ID of the stored message. Required. Valid values are numbers ≥ 0 .

Output from Service to Business Process

The following table contains the parameter passed from the Mailbox Extract Commit service to a business process:

Parameter	Description
MessageId	ID of the stored message.

Mailbox List Service

The following table provides an overview of the Mailbox List service:

System name	Mailbox List Service
Graphical Process Modeler (GPM) category	None
Description	Lists all mailbox paths that the user has permission to use. If a virtual root is configured for the user, the mailbox paths are returned as relative paths.
Business usage	This is a system service.
Usage example	The Mailbox Browser Interface uses this service to display the mailbox path of a calling user.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox List service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Query service – Enables querying messages.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	Not applicable (Internal service)
Business process context considerations	No

Returned status values	The possible status values a service can return are: <ul style="list-style-type: none">◆ Success – Normal completion.◆ Error – A generic error occurred.
Restrictions	This is a system service and is not intended for use in business processes.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox List Service Works

The Mailbox Browser Interface (MBI) uses the Mailbox List service to display all mailboxes a user has permission to use.

Mailbox Query Service

The following table provides an overview of the Mailbox Query service:

System name	Mailbox Query Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > Mailbox
Description	Performs a query of Application mailboxes for messages that meet specified criteria and returns results.
Business usage	Use this service to find messages in Application Mailbox.
Usage example	Search for messages in one or more Application mailboxes.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>The Mailbox Query service works with the other Application Mailbox services to provide mailboxing capability:</p> <ul style="list-style-type: none">◆ Mailbox Add service – Enables the insertion of messages into a mailbox.◆ Mailbox Extract Begin service – Enables the extraction of messages from a mailbox.◆ Mailbox Extract Commit service – Provides the ability for a business process to signal Application Mailbox that the message extract request has successfully completed.◆ Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Application Mailbox.◆ Mailbox Delete service – Enables deletion of messages.◆ Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Application Mailbox.◆ Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	A business process using this service must be run by a user with permission to access all applicable mailboxes. A user with Mailbox Global Query permission can query any message in any mailbox.
Business process context considerations	The service extracts the user permissions at run time to confirm authorization of the business process to access the referenced mailboxes.

Returned status values	<p>Possible status values:</p> <ul style="list-style-type: none"> ◆ Success – Normal completion. ◆ User Permission Error – The user associated with the business process either does not have Mailbox Global Query permission, or does not have permission to use the target mailboxes. ◆ Invalid Mailbox Error – The mailbox identified as the query target does not exist. ◆ Invalid Mailbox Parameter Error – An error occurred passing parameters to this service such as a message ID in an invalid format. ◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred. ◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.
Restrictions	The limitations of this service are based on the assignment of mailbox permissions to users and groups. A Mailbox Global Query permission allows access to all mailboxes.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Query Service Works

Use the Mailbox Query service to query Application Mailbox and find messages.

Business Process Example

The following BPML queries all messages currently in the Abcd mailbox. If the user has a virtual mailbox root, then the Abcd mailbox path is relative to the user's virtual mailbox root.

The user must have permission for the Abcd Mailbox, or have the Mailbox Global Query permission.

```
<process name="AbcdQuery">
  <sequence name="Query">
    <!-- Query the Mailbox -->
      <operation name="Mailbox Query Service">
        <participant name="MailboxQuery"/>
        <output message="QueryRequest">
          <assign to="." from="*"></assign>
          <assign to="MailboxPath">/Abcd</assign>
        </output>
        <input message="inmsg">
          <assign to="QueryResults" from="*"></assign>
        </input>
      </operation>
    </sequence>
  </process>
```

Implementing the Mailbox Query Service

To implement the Mailbox Query service for use in a business process:

1. Create an Mailbox Query service configuration. For information, see *Managing Services and Adapters*.

2. Configure the Mailbox Query service. For information, see *Configuring the Mailbox Query Service* on page 941.
3. Use the Mailbox Query service in a business process.

Configuring the Mailbox Query Service

The following table describes the fields used to configure the Mailbox Query service in the GPM:

Field	Description
Config	Name of the adapter configuration.
Ascending	Whether the business process should sort the result of a query in ascending order. Valid values are Yes and No.
EndDateTime	Defines the latest value for this query. Valid values are any date and time in the following format: Year-Month-Day Hour:Minute:Second (yyyy-mm-dd hh:mm:ss) For example, 2006-02-21 04:02:20
MailboxPath	One or more mailbox paths that the user has permission to use. If the user has been set up with a virtual root, then MailboxPath will be a relative path to the virtual root. The virtual root is not visible to the business process. Required if you do not have Mailbox Global Query Permission. Valid values are UNIX-style paths where the folders correspond to a mailbox hierarchy. Paths must begin with the '/' character and use the '/' to delimit mailboxes in the hierarchy. The space character is allowed in the middle of a mailbox name. The following characters are <u>not allowed</u> anywhere in a mailbox name: ? < > " : \ / % and *.
MessageExtractable	Constraint to query messages based on their Extractability: <ul style="list-style-type: none"> ◆ Yes – Either ExtractableCount >0, ExtractableUntil is current date or later, or Extractable = Yes. ◆ No – Either ExtractableCount = 0, ExtractableUntil is an earlier date, or Extractable = No.
MessageId	ID of the stored message. Required. Valid values are numbers greater than or equal to 0.
MessageNamePattern	Pattern to select messages with names matching the pattern. The following characters cannot be used in a message name: ? < > / " % * You can use an asterisk (*) as a wildcard such as *.po.
OrderBy	Enables the business process to sort the result of a query by any of the following: <ul style="list-style-type: none"> ◆ Message Path ◆ Message Size ◆ Message ID ◆ Mailbox Path ◆ Created Date and Time
StartDateTime	Defines the earliest value for this query. Valid values are any date and time in the following format: Year-Month-Day Hour:Minute:Second (yyyy-mm-dd hh:mm:ss) For example, 2006-02-21 04:02:20

Field	Description
User ID	User ID that added the message.

Parameters Passed from a Business Process to the Service

The following table contains the parameters passed from the business process to the Mailbox Query service:

Parameter	Description
MailboxPath	One or more mailbox paths that the user has permission to use. If the user has a virtual root, then MailboxPath is a relative path to the virtual root. The virtual root is not visible to the business process. Required if you do not have Mailbox Global Query Permission. Valid values are UNIX-style paths where the folders correspond to a mailbox hierarchy. Paths must begin with the '/' character and use the '/' to delimit mailboxes in the hierarchy. The space character is allowed in the mailbox name. The following characters are <u>not allowed</u> anywhere in a mailbox name: ? < > " : \ / % and *.
MessageNamePattern	Pattern to select messages with names matching the pattern. The following characters cannot be used in a message name: ? < > / " % * You can use an asterisk (*) as a wildcard such as *.po.
StartDateTime	Defines the earliest value for this query. Valid values are any date and time in the following format: Year-Month-Day Hour:Minute:Second (yyyy-mm-dd hh:mm:ss) For example, 2006-02-21 04:02:20
EndDateTime	Defines the latest value for this query. Valid values are any date and time in the following format: Year-Month-Day Hour:Minute:Second (yyyy-mm-dd hh:mm:ss) For example, 2006-02-21 04:02:20
DocumentId	A valid document ID for an existing document in Application.
MessageExtractable	Constraint to query messages based on their Extractability: <ul style="list-style-type: none"> ◆ Yes – Either ExtractableCount >0, ExtractableUntil is current date or later, or Extractable = Yes. ◆ No – Either ExtractableCount = 0, ExtractableUntil is an earlier date, or Extractable = No.
QueryStartPos	Specifies the starting position value for the query results. Use to pull a portion of the query results when there are a large number of messages. Use in conjunction with QueryEndPos to determine how many page links to display, and the size of the result set. Valid value is any positive integer. Optional.
QueryEndPos	Specifies the ending position value for the query results. Use to pull a portion of the query results when there are a large number of messages. Use in conjunction with QueryStartPos to determine how many page links to display, and the size of the result set. Valid value is any positive integer. Optional.
GetVisibilityData	Get message metadata including visibility information. Valid values are Yes or No. Optional. <ul style="list-style-type: none"> ◆ Yes - Will return the following parameters: AddedByProtocol, AddedByPrincipal, AddedByIP, AddedByWorkflowID, AddSecure, AvailableExtractCount, Extracted, and ExtractAttempt.

Parameter	Description
RecentExtractLimit	Specifies the maximum number of extracts to return based on the most recent communications sessions. Valid value 1 – 100,000. Optional.

Parameters Passed from Service to Business Process

The following table contains the parameters passed from the Mailbox Query service to a business process:

Parameter	Description
MessageId	ID of the stored message.
DocumentId	Document ID corresponding to the stored message.
CreateDateTime	Creation date. Format <i>yyyyMMdd'T'hhmm</i> .
MessageName	Name of the message.
MailboxPath	Path of the mailbox which the message was added to. If the user has been set up with a virtual root, the MailboxPath will be a relative path to the virtual root. The virtual root is not visible to the business process.
ContentType	Indicates the MIME type and subtype. Use the following format: <i>MIME Type/MIME Subtype</i> .
MessageSize	Size of the added message in bytes.
ExtractableCount	Number of times this message may be accessed. One of the three extractability parameters is returned.
Extractable	Indicates whether this message can be extracted.
ExtractableUntil	The last date and time that this message may be extracted.
QueryTotal	The total number of messages in the query. Based on the values of QueryStartPos and QueryEndPos, the service returns a limited result set. If QueryStartPos and QueryEndPos are specified, the QueryTotal output is returned.
ContentEncrypted	Indicates whether or not the message content is encrypted. Valid values are True or False.
AddedByProtocol	Indicates whether or not the message was added by a protocol. Valid values are SFTP, FTP, WebDAV, CDInterop, HTTP, MailboxService, MBI, or AS2. Returned when GetVisibilityData=Yes.
AddedByPrincipal	Indicates if the message was added by the Application user. Valid values include any possible User ID. Returned when GetVisibilityData=Yes.
AddedByIP	Specifies the IP address of the user responsible for adding the message. Returned when GetVisibilityData=Yes.
AddedByWorkflowID	Specifies the ID of the Business Process responsible for adding the message. Returned when GetVisibilityData=Yes.
AddSecure	Indicates whether or not the message is encrypted or was added with a protocol configured for SSL. Valid values are True or False. Returned when GetVisibilityData=Yes.
CurrentlyExtractable	Based on the extractability policy; indicates whether or not the message is currently extractable.

Parameter	Description
AvailableExtractCount	From the available recorded extracts; indicates the number of times the message was successfully extracted. Returned when GetVisibilityData=Yes.
Extracted	A set of recent extract attempts. Returned when GetVisibilityData=Yes.
ExtractAttempt	A recent extract attempt. Returned when GetVisibilityData=Yes.
Principal	Indicates the user who performed the extract. Subelement of ExtractAttempt.
IP	Specifies the IP address from which the extract was performed. Subelement of ExtractAttempt.
Protocol	Specifies the protocol used to perform the extract. Valid values are SFTP, FTP, WebDAV, CDInterop, HTTP, MailboxService, MBI, or AS2. Subelement of ExtractAttempt.
Date	Extract attempt date. Format <i>yyyyMMdd'T'hhmm</i> . Subelement of ExtractAttempt.
Status	Indicates whether or not the extract attempt was successful. Subelement of ExtractAttempt.
Secure	Indicates whether or not the extract attempt was secure. Valid values are True or False. Subelement of ExtractAttempt.
Binary Transfer	Indicates whether or not the extract attempt was performed in binary mode. (For FTP protocol.) Subelement of ExtractAttempt.
WorkflowID	Specifies the ID of the Business Process responsible for the extract. Subelement of ExtractAttempt.

Sample Visibility Data

The GetVisibilityData parameter allows you to retrieve extra information about a message. For example, consider the following input parameters:

```
<attribute name="GetVisibilityData">Yes</attribute>
<attribute name="RecentExtractLimit">10</attribute>
```

Based on these settings, the following output is returned in addition to the standard message details.

```
<AddedByIP>192.168.100.1</AddedByIP>
<AddedByPrincipal>Bob</AddedByPrincipal>
<AddedByProtocol>FTP</AddedByProtocol>
<AddedByWorkflowId>-1</AddedByWorkflowId>
<AddSecure>>false</AddSecure>
<AvailableExtractCount>1</AvailableExtractCount>
<Extracted>
  <ExtractAttempt>
    <Principal>Jim</Principal>
    <IP>192.168.100.5</IP>
    <Protocol>SFTP</Protocol>
    <Date>01/01/2001</Date>
    <Status>success</Status>
    <Secure>>true</Secure>
    <BinaryTransfer>>true</BinaryTransfer>
    <WorkflowId>-1</WorkflowId>
  </ExtractAttempt>
</Extracted>
```

Mailbox Scheduled Delete Service

The following table provides an overview of the Mailbox Scheduled Delete service:

System name	Mailbox Scheduled Delete Service
Graphical Process Modeler (GPM) category	None
Description	Schedules message deletion in one or many mailboxes.
Business usage	Use this service to create a schedule to delete mailbox messages that meet specific criteria.
Usage example	The Application Mailbox system administrator schedules the deletion of messages older than 30 days on Sundays at 2:00 a.m.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	The business process using this service must be run by a user with permission to use any and all mailboxes that the delete request will attempt to delete. The exception to this rule is if the user has the Mailbox Global Delete permission. With the Mailbox Global Delete permission, this service will enable message deletion across all mailboxes.
Business process context considerations	The service extracts the user permissions at run time to confirm authorization of the business process to access the referenced mailboxes.
Returned status values	Possible status values: <ul style="list-style-type: none">◆ Success – Normal completion.◆ User Permission Error – The user associated with the business process either does not have Mailbox Global Delete permission, or does not have permission to use the target mailbox.◆ Invalid Mailbox Error – The mailbox identified as the delete target does not exist.◆ Invalid Mailbox Parameter Error – An error occurred passing parameters to this service such as a message ID in an invalid format.◆ Mailbox Repository Error – A generic error associated with the mailbox repository has occurred.◆ Mailbox Service Error – A generic error associated with the mailbox service has occurred.

Restrictions	The limitations of this service are based on the assignment of mailbox permissions to users and groups. This service also can be used by users assigned the Mailbox Global Delete permission.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Scheduled Delete Service Works

Use the Mailbox Scheduled Delete service to delete messages from one or more mailboxes. As with all Application scheduled services, you can configure the scheduled delete service to execute once at a specified date and time, or periodically such as once a month or twice a week.

Implementing the Mailbox Scheduled Delete Service


To implement the Mailbox Scheduled Delete service for use in a business process:

1. Create an Mailbox Scheduled Delete service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Scheduled Delete service. For information, see *Configuring the Mailbox Scheduled Delete Service* on page 946.
3. Use the Mailbox Scheduled Delete service in a business process.

Configuring the Mailbox Scheduled Delete Service

To configure the Mailbox Scheduled Delete service, you must specify settings for the following fields in the Application Admin Console or GPM:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Delete only messages older than (days)	The number of days a message can exist in Application Mailbox before being deleted. You can use any number greater than zero. Required.

Field	Description
Message Extractability	<p>Delete only unextractable messages – Only messages that are no longer extractable will be deleted. This includes with the following extract policy values:</p> <ul style="list-style-type: none"> ◆ ExtractableCount = 0 ◆ ExtractableUntil is an earlier date or Extractable = No <p>Delete All messages – All messages (extractable and unextractable) are deleted.</p>
Message Name Pattern	<p>Pattern to select messages with names matching the pattern. The following characters cannot be used in a message name \ / : ? " < > </p>
Select Mailboxes for Message Removal	<p>Mailbox paths to be queried during message deletion. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ All Mailboxes – Includes all mailboxes ◆ Selected Mailboxes – Displays a mailbox picker to enable you to choose specific mailboxes for the query <p>Note: Leaving this field blank (by not selecting All or Selected Mailboxes) will cause errors to be written to the status report.</p>
Run as User	<p>Applies to the scheduling of the business process. The Run As User field only displays as an option if <i>Does this service start a business process?</i> is set to Yes.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid value is any valid Application user ID.</p> <p>Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>
Use 24 Hour Clock Display	<p>If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock.</p>

Field	Description
Schedule	<p>Information about scheduling the business process invoked by this service.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, the service does not start a business process and does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Output from Service to Business Process

The following table contains the parameter passed from the Mailbox Scheduled Delete service to a business process:

Parameter	Description
MessageId	ID of the stored message.

Mailbox Update Service

The following table provides an overview of the Mailbox Update service:

System name	Mailbox Update Service
Graphical Process Modeler (GPM) category	None
Description	Updates messages in a mailbox.
Business usage	A trading partner would use this service to update messages in Application Mailbox.
Usage example	A business process updates the extractability of a message, or resubmits the message for routing.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	Nothing external to Application is required to use this service.
Initiates business processes?	No
Invocation	Not applicable (Internal service)
Business process context considerations	No
Returned status values	Possible status values: <ul style="list-style-type: none">◆ Success – Normal completion.◆ User Permission Error – The user associated with the business process does not have permission to use the target mailbox.◆ Invalid Mailbox Error – The mailbox identified as the query target does not exist.◆ Mailbox Message Not Found Error – The message specified cannot be found.◆ Mailbox Message Multiple Extractable Types Specified By Business Process Error – The data sent to the process contains specifications for more than one extractability type.◆ Mandatory Parameter Missing Error – A necessary parameter (such as MessageId) is missing.◆ Mailbox Repository Error – A generic error associated with the mailbox repository occurred.◆ Mailbox Service Error – A generic error associated with the mailbox service occurred.

Restrictions	The limitations of this service are based on the assignment of mailbox permissions to users and groups.
Testing considerations	Troubleshooting information for this service can be found in Application Mailbox log files.

How the Mailbox Update Service Works

The Mailbox Update service is used to update the status of a message.

Implementing the Mailbox Update Service

To implement the Mailbox Update service for use in a business process:

1. Create an Mailbox Update service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Mailbox Update service. For information, see *How the Mailbox Update Service Works* on page 950.
3. Use the Mailbox Update service in a business process.

Configuring the Mailbox Update Service

To configure the Mailbox Update service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the adapter configuration.
Extractable	Indicates whether this message can be extracted. Valid values are Yes and No. Note: You must provide a value for this field or one of the following: <ul style="list-style-type: none"> ◆ ExtractableCount ◆ ExtractableUntil
ExtractableCount	Number of times this message may be accessed. Optional. One of the three extractability parameters is returned. Valid values are ≥ 0 . Note: You must provide a value for this field or one of the following: <ul style="list-style-type: none"> ◆ Extractable ◆ ExtractableUntil
ExtractableUntil	The last date and time that this message may be extracted. Valid values are dates and times in <i>yyyyMMddThhmm</i> format. Note: You must provide a value for this field or one of the following: <ul style="list-style-type: none"> ◆ Extractable ◆ ExtractableCount
MessageId	ID of the stored message. Required. Valid values are numbers ≥ 0 .

Field	Description
ResubmitMessage	Indicates if this message should be resubmitted for routing. Valid values are Yes and No.

Map Test Service

The Map Test service runs only in concert with the Map Editor to enable you to remotely test a compiled map (.txo file) from a client machine prior to checking the map in to the Application server.

The following table provides an overview of the Map Test service:

System Name	MapTestService
Graphical Process Modeler (GPM) categories)	All Services
Description	This service accepts a message containing a compiled map and associated data, runs the data through translation, and returns a translation report and output data to the user.
Business usage	This service is used by the Map Test feature of the Map Editor to allow testing of a map from the client machine without having to check the map into Application. Note: Although you can run this service from the client, the actual map is executed on the Application Server. We recommend testing maps only against Test of Development systems, not against a Application production system.
Usage example	A User runs the Map Test feature to remotely test a compiled map (.txo) from a client machine prior to checking the map in to the Application server, .to ensure that the map translates data correctly and efficiently, prior to executing the map in a Production environment.
Preconfigured?	Yes
Requires third party files?	None
Platform availability	All supported Application platforms.
Related services	SOAP service, HTTP Server adapter, Translation service
Application requirements	None
Initiates business processes?	No
Invocation	Used internally Application; invoked by the Map Editor Map Test feature on the client.
Business process context considerations	None
Returned status values	Success Error
Restrictions	If you have not configured your operating system to specify default programs to open .txt and .xml files, the test result files may not be automatically displayed. If this is the case, however, you can locate and open the files using the file/path location.
Persistence level	System default
Testing considerations	None

How the Map Test Service Works

When you use the Map Test service, the Map Editor takes a compiled map (translation object) and a data file to run with the map, and loads both the translation object and the data file into an XML SOAP message. When the Map Test service runs, it is visible in the Application Current Processes interface.

Using HTTP, the Map Test client posts the XML SOAP message to the Map Test service. The Map Test service submits the SOAP message to the SOAP service (inbound or outbound), and the SOAP service disassembles the message and returns the translation object and associated data back to the Map Test service.

Note: You need to enable or start the Map Test service; by default it is disabled.

The Map Test service then submits the translation object and the data to the Translation service, which runs translation using the supplied translation object and data, and returns the output data and a translation report to the Map Test service.

The Map Test service loads the translation report and output data into another XML SOAP message and sends it to the client using the HTTP Server adapter. Then the Map Editor disassembles the SOAP message and presents the user with the translation report (in XML format) and the output translation data. If there is no translation report, the Map Editor returns a file stating that no translation report is available.

Implementing the Map Test Service

Note: You do not need to create a configuration of the Map Test service. However, since the default is for the service to be disabled, you do need to enable it to use the Map Test feature. That is, in the **translator.properties** file the **maptest.MaptestServiceEnabled** property is set to False by default. If you do not set the value to True using the Customer Override feature (explained below), the service will not accept any map test requests.

You can turn off the Map Test service to prevent users who have access to a trading partner's Application system from attempting to use the Map Test feature to run translation on that trading partner's system. Turning off the Map Test service prevents the possible execution of JDBC maps which could access production data. Application supports the use of a customer override property file to override property settings in the property files. The customer override property file is not changed during installation of Application upgrades or patches. To prevent having your customized settings overwritten, you should use the customer override property file whenever possible rather than editing the Application property files directly.

To enable the Map Test service, complete the following tasks:

Note: If the Map Test service is disabled by the **translator.properties** entry, an error message returned to the client and is presented to the user in a format like the translator report.

1. In the *install_dir*/properties directory, locate (or create, if necessary) the **customer_overrides.properties** file.
2. Open the **customer_overrides.properties** file in a text editor.
3. Add the property you want to override, using the following format:
`translator.maptest.MaptestServiceEnabled=true`
4. Save and close the **customer_overrides.properties** file.

5. Stop and restart Application to use the new values.
6. Test your changes to ensure that the overrides give the desired results. If you have problems, contact Sterling Commerce Customer Support for assistance.

Mapped Extraction Service

The Mapped Extraction service enables you to save data generated within Application processing in a Visibility Services fact repository. The following table provides an overview of the Mapped Extraction service:

System Name	Mapped Extraction Service
Graphical Process Modeler (GPM) categories	All Services
Description	Use the Mapped Extraction service in a business process to specify a fact model and a mapped extraction name. You enter the name of the map in the mapped extraction element in the fact model itself. See <i>Extracting Data</i> for more information.
Business usage	This service enables you to save data generated within Application processing in a Visibility Services fact repository
Usage example	Business data that is generated during Application processing can be specified as having a particular use for gathering business intelligence.
Preconfigured?	Yes. There is a configuration of the service called MappedExtractionService that is installed with Application.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Straight Through Extraction service
Application requirements	The Visibility Services fact model referenced by this service (and the translation map, if used), must be checked in to Application before running the business process.
Initiates business processes?	Cannot initiate a business process.
Invocation	From a business process
Business process context considerations	You must supply the name of the fact model and the mapped extraction element.
Returned status values	<ul style="list-style-type: none">◆ Success – The service gets the required parameters and sends event successfully.◆ Error – Failed to process the data, the required parameters do not exist.
Restrictions	None
Testing considerations	Ensure that the Visibility Services listeners are running. See <i>About Event Listeners</i> for more information.

How the Mapped Extraction Service Works

1. At run time, the Mapped Extraction service fires a mapped extraction event (Event Schema key: “BI.Event.Mapped”), using the specified mapped extraction name and fact model name.
2. The primary document object ID is sent in the same event.
3. The Visibility Services mapped extraction event listener picks up the event, translates the primary document, and puts data into the fact repository.

Using Pass Through Mode

If you are not translating the data as part of the extraction, you can specify “passThrough” as the mapped extraction name in the fact model. Your data must comply with the format of the fact model schema (*factmodelname.xsd*). See *Extracting Data* for more information about using passThrough.

Implementing the Mapped Extraction Service

Use the supplied configuration of the service, MappedExtractionService, in your business processes.

The only configuration required for the service is specifying the fact model and mapped extraction to be used for your business process. You specify these parameters in the GPM.

Field	Description
Config	Select the name of the service configuration from the list. You can use the supplied configuration, MappedExtractionService.
factmodel	Enter the name of the fact model that contains the mapped extraction element from the previous field. Required. Case sensitive. The name must be entered exactly as it appears in the fact model.
mappedextraction	Enter the name of the Mapped Extraction element that should be referred to by this business process. Required. Case sensitive. The name must be entered exactly as it appears in the fact model. This is the mapped extraction element that contains the name of the translation map, if using a map in this extraction.

Business Process Example

The following example calls the predefined service MappedExtraction with mapped extraction name and fact model name:

```
<process name="demoMappedExtraction">
  <sequence name="main seq">
    <operation name="MappedExtraction">
      <participant name="MappedExtraction"/>
      <output message="MappedExtractionInputMessage">
        <assign to="mappedextraction">demoBIMappedExtraction</assign>
        <assign to="factmodel">BankFactModel</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

```
</operation>  
</sequence>  
</process>
```

MergeDocument Service

The MergeDocument service merges two documents into a single document and places it into the primary document. The following table provides an overview of the MergeDocument service:

System Name	MergeDocument
Graphical Process Modeler (GPM) categories)	All Services
Description	The MergeDocument service merges two documents into a single document and places it into the primary document.
Business usage	A business user would use this service to merge two existing and separate documents into one document.
Usage example	A Application business process uses the file system adapter to collect a file. The collected file is first placed into the primary document, and then assigned to a document key to make it available in the document area. The file system adapter collects another file and it, too, is ultimately placed in the document area. The MergeDocument service is then used to combine the two documents from the document area into one, and the resulting merged document is placed into the primary document.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	This service is initiated from a business process.
Business process context considerations	Usage of this service requires a document key to be created from the primary document. See the example business process for more details.
Returned status values	0 – Success 1 – Error
Restrictions	None
Persistence level	Default
Testing considerations	None

How the MergeDocument Service Works

Two documents are passed to the MergeDocument service. The service merges the documents into a single document and places the resulting document in the primary document. For example, if document1 (FirstPriDoc) contains the text “This is test document 1,” and document2 (SecondPriDoc) contains the text “This is test document 2,” then the new primary document (after the merge) will contain the text “This is test document 1This is test document 2.”

The MergeDocument service is preconfigured, so there is no implementation or configuration necessary.

Parameters Passed From the Business Process to the Service

The following table contains the parameters passed from the business process to the MergeDocument service:

Field	Description
document1	Specify the path and file name of a document that you want to merge with another document.
document2	Specify the path and file name of the document that you want to merge with the document specified in document1 .

Business Process Example

The following example illustrates how the MergeDocument service can be used in a business process:

```
<process name="MergeDocumentProcess">
  <sequence>
    <operation name="File System Adapter">
      <participant name="FileSystemAdapterInstance"/>
      <output message="FileSystemInputMessage">
        <assign to="." from="*"></assign>
        <assign to="Action">FS_COLLECT</assign>
        <assign to="deleteAfterCollect">>false</assign>
        <assign to="filter">file1.txt</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <assign to="doc1" from="PrimaryDocument/@SCIOBJECTID"></assign>
    <operation name="File System Adapter">
      <participant name="FileSystemAdapterInstance"/>
      <output message="FileSystemInputMessage">
        <assign to="." from="*"></assign>
        <assign to="Action">FS_COLLECT</assign>
        <assign to="deleteAfterCollect">>false</assign>
        <assign to="filter">file2.txt</assign>
      </output>
      <input message="inmsg">
```

```

        <assign to="." from="*"></assign>
    </input>
</operation>

<assign to="doc2" from="PrimaryDocument/@SCIOBJECTID"></assign>
<operation name="MergeDocument">
    <participant name="MergeDocumentInstance"/>
    <output message="MergeDocumentInputMessage">
        <assign to="." from="*"></assign>
        <assign to="document1" from="'doc1'"></assign>
        <assign to="document2" from="'doc2'"></assign>
    </output>
    <input message="inmsg">
        <assign to="." from="*"></assign>
    </input>
</operation>

</sequence>
</process>

```

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the MergeDocument service to the business process:

Parameter	Description
PrimaryDocument	The link to the primary document. Optional.

MIME Service

Use the MIME service to build (construct) or decode an RFC822-compliant MIME multipart document. The following table provides an overview of the MIME service:

System name	MIME Service
Graphical Process Modeler (GPM) categories	All Services, Communications
Description	Encodes and decodes MIME multipart messages.
Business usage	Performs two functions: <ul style="list-style-type: none">◆ Builds (constructs) a MIME multipart message from an XML representation of a MIME message.◆ Decodes a MIME multipart message into its corresponding XML representation. Use the MIME service if an RFC 822-compliant document is needed.
Usage example	<ul style="list-style-type: none">◆ As part of a business process, you want to create a MIME multipart document to be sent using any of the transport adapters.◆ You want a business process to parse an incoming MIME multipart document.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	None
Restrictions	The XML node to be MIME encoded must be created in the expected XML representation. The construction of the MIME multipart document is dependent on the structure of the XML node that is being passed to the service. The service might fail if the XML node does not follow the correct structure. See <i>Business Scenario</i> on page 962 and <i>Usage Examples</i> on page 969 for information about how to create a valid XML node representation.

Testing considerations	The easiest way to test this service is to set up a business process that reads in the files to be MIME encoded, creates the XML representation of the MIME message, and then runs the MIME service to encode the documents. You can then decode the MIME document and make sure it is represented in the format you expect. See <i>Business Scenario</i> on page 962 and <i>Usage Examples</i> on page 969 for information.
------------------------	--

How the MIME Service Works

The MIME service is used to encode or decode an RFC822-compliant MIME multipart document. An RFC822-compliant MIME multipart document is a MIME document that includes one or more attachments and has the sender and recipient addresses, as well as the subject header, written in the headers. The attachments can be of different file types, for example, a Microsoft Word document or Excel spreadsheet.

The following section describes a business scenario in which you can use the MIME service, along with a sample solution.

Business Scenario

Your company exchanges product information with a trading partner. Your trading partner expects RFC822-compliant MIME documents from you. You need to send one Microsoft Word file and one text file as attachments. After you create the RFC822-compliant MIME document, you need to make it the primary document so another business process can send it to your trading partner.

Business Solution Example

The following approach is used to solve the above business scenario.

1. Use the pre-configured MIME service or create your own MIME service instance.
2. Create a business process that does the following:
 - a. Uses the File System adapter to read in the documents that need to be encoded.
 - b. Sets up an XML representation of the MIME message in the process data of the business process and adds the documents to the XML in process data. For more information, see *Process Data Prerequisites* on page 962.
 - c. The MIME service encodes the documents and assigns the MIME-encoded document as the primary document.
 - d. Sends the MIME-encoded document to the trading partner using the SMTP Send adapter. For more information, see *SMTP Send Adapter*.

Note: The example in this section focuses only on the MIME service functionality.

Process Data Prerequisites

To use the MIME service to MIME-encode documents, you need to construct an XML representation of the MIME message and place it in the process data of the business process prior to calling the MIME service. The upper-level mime:message element placed in process data is input to the MIME service. See *Business Process Modeling Language (BPML) Example* on page 963.

The XML representation of the MIME message contains three main elements that represent a MIME part. These elements include:

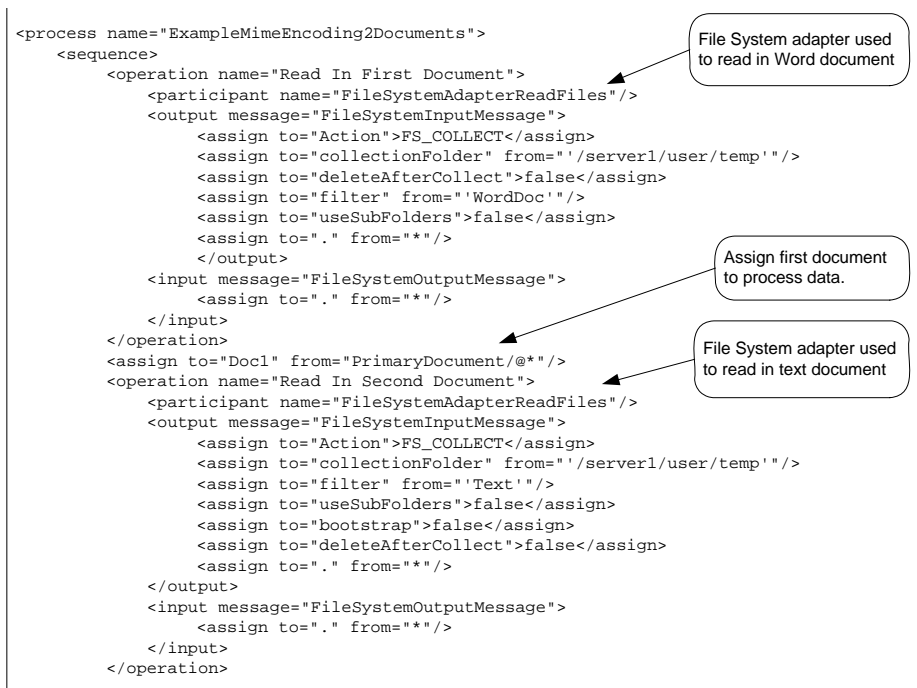
- mime:message – This element is the root node for the MIME part.
- mime:header – This element creates the specific header for the MIME part.
- mime:body – This element creates the content of the MIME part.

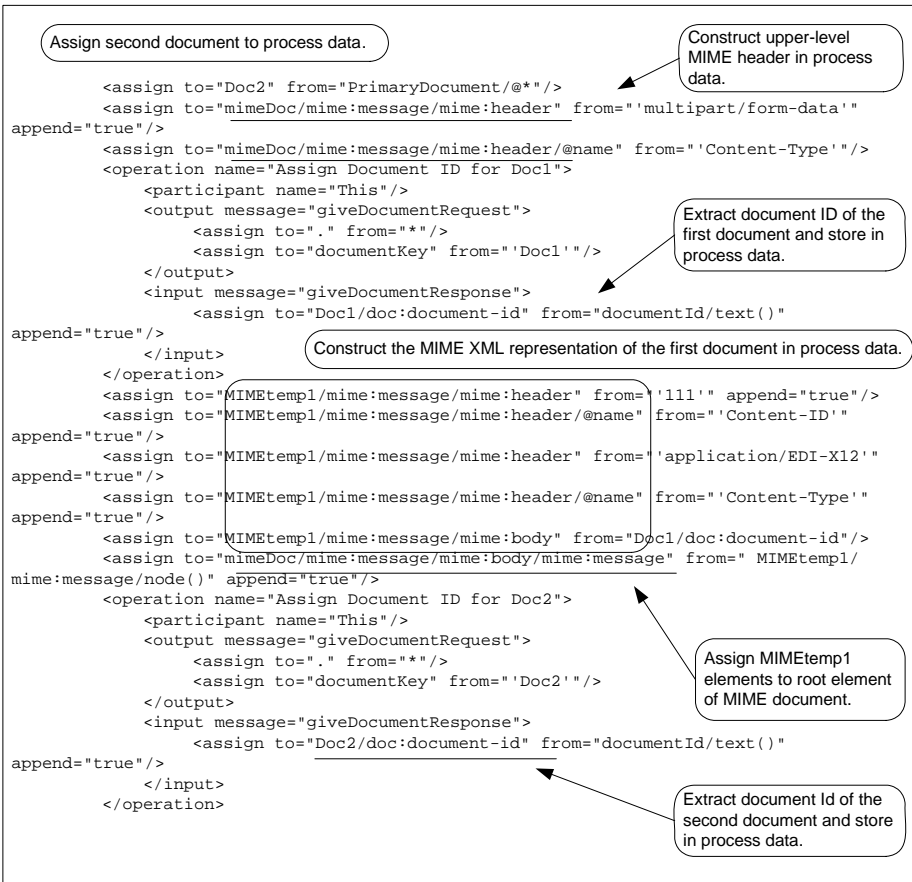
Because our example is a multipart message that contains two attachments, each MIME part has its own individual headers and body. To support this format, you need to create additional mime:message, mime:header, and mime:body elements underneath the first mime:body element for each of your attachments.

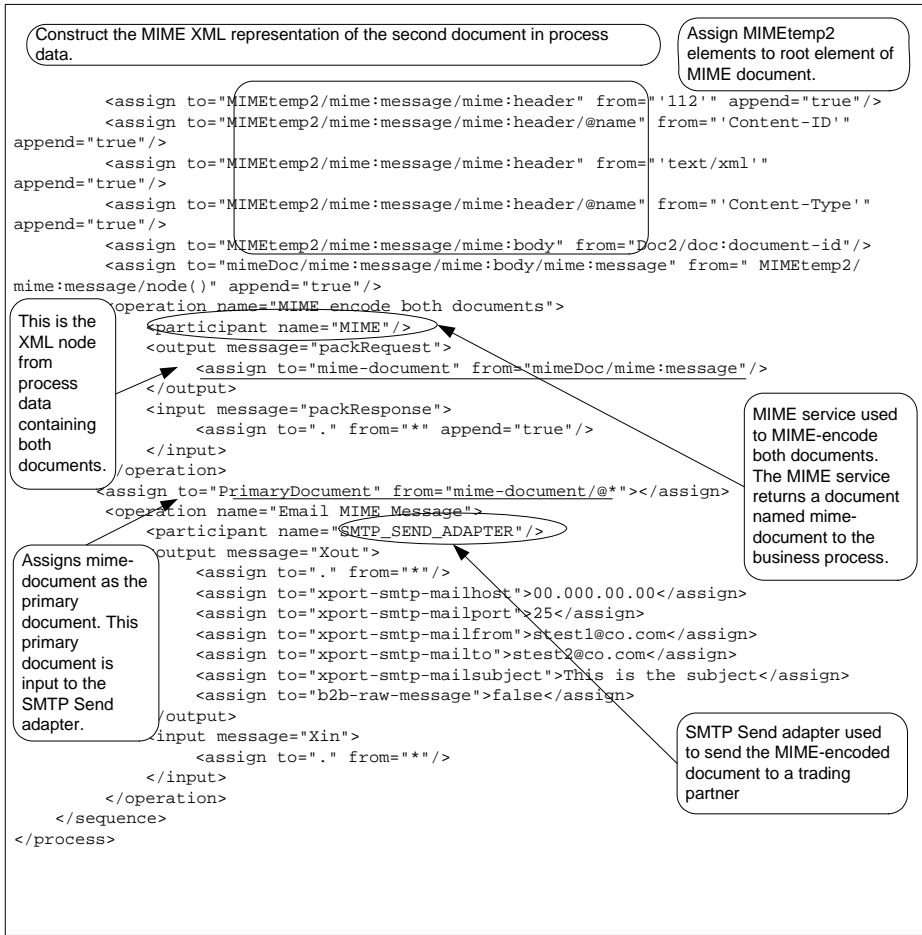
The example in the following section illustrates this point. In addition, see *Setting up Process Data for MIME Encoding* on page 969.

Business Process Modeling Language (BPML) Example

The following example shows a solution to the business scenario using BPML.







The following example shows the input to the MIME service from process data. This XML representation of the MIME message was created in the above business process prior to starting the MIME service. The MIME service processes everything under the first-level mime:message element (see the output message defined for the MIME service in the above example).

```

<mimeDoc>
  <mime:message xmlns:mime="http://www.company.com/company-name/mime/v0.5">
    <mime:header name="Content-Type">multipart/form-data</mime:header>
    <mime:body>
      <mime:message>
        <mime:header name="Content-ID">111</mime:header>
        <mime:header name="Content-Type">application/EDI-X12</mime:header>
        <mime:body>
          <doc:document-id
xmlns:doc="http://www.company.com/company-name/document-id">serverName:76f6d6a8:fbbb
18bfda:-6b01</doc:document-id>
        </mime:body>
      </mime:message>
      <mime:message>
        <mime:header name="Content-ID">112</mime:header>
        <mime:header name="Content-Type">text/xml</mime:header>
        <mime:body>

```

```

<doc:document-id xmlns:doc="http://www.company.com/company-name/document-id">
serverName:76f6d6a8:fbbb18bfda:-6aea</doc:document-id>
</mime:body>
</mime:message>
</mime:body>
</mime:message>
</mimeDoc>

```

The following example shows the output returned from the MIME service to the business process. This output is returned as a document named mime-document. The business process explicitly assigns the mime-document to primary document (see the Assign statement following the MIME service in the previous BPML), which is then passed to the SMTP Send adapter.

```

Content-Type: multipart/form-data; boundary="-server1::76f6d6a8:fbbb18bfda:-6331::multipart.boundary.string"
Content-Length: 19757

--serverName::76f6d6a8:fbbb18bfda:-6331::multipart.boundary.string
Content-ID: 111
Content-Type: application/EDI-X12
_Microsoft Word Document
MSWordDoc_Word.Document.89q
-- serverName::76f6d6a8:fbbb18bfda:-6331::multipart.boundary.string
Content-ID: 112
Content-Type: text/xml

Hello World

-- serverName::76f6d6a8:fbbb18bfda:-6331::multipart.boundary.string

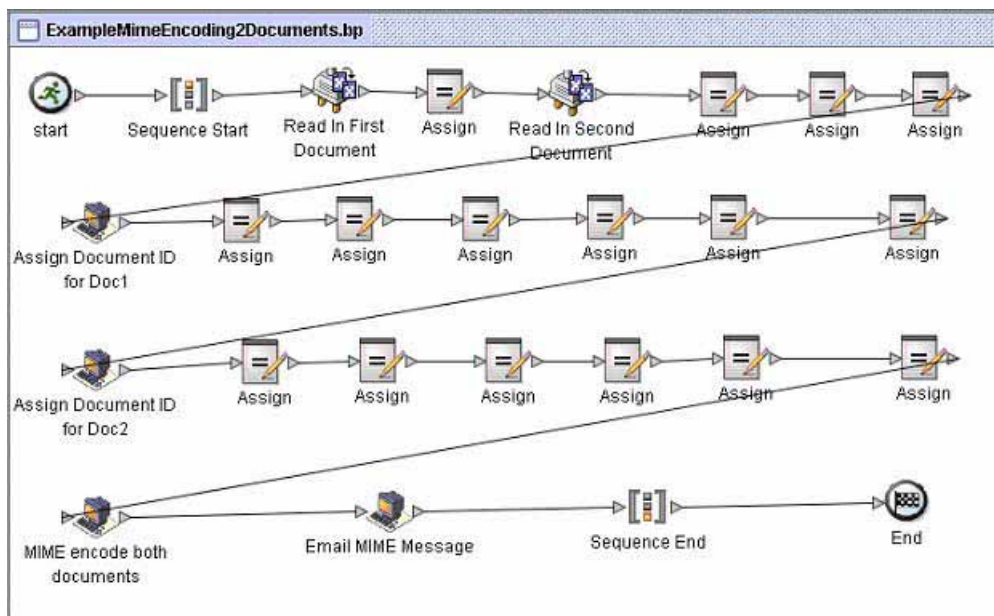
```

Multipart for MS Word document

Multipart for text document

GPM Example

The following example shows the corresponding GPM view of the business process.



For additional MIME service examples, see *Usage Examples* on page 969.

Implementing the MIME Service

You can implement the MIME service in two ways:

- To MIME-encode documents
- To MIME-decode documents

The information in this section applies to both implementations.

To implement the Mime service for use in a business process:

1. Collect the following information:

For MIME-encoding:

- ◆ The documents you want to MIME encode
- ◆ The XML representation of the MIME message for the MIME service to use (See *Setting up Process Data for MIME Encoding* on page 969 for more information.)

For MIME-decoding:

- ◆ The document ID of the MIME-encoded document
2. Create a Mime service configuration. For information, see *Managing Services and Adapters*.
 3. Configure the Mime service. For information, see *Configuring the Mime Service* on page 967.
 4. Use the Mime service in a business process.

Configuring the Mime Service

To configure the Mime service, you must specify settings for the following fields in the GPM:

Name	Value	Use XPath?
debug		<input type="checkbox"/>
inline-first-body-part		<input type="checkbox"/>
mime-document	nodeInProcessData/mime:message	<input checked="" type="checkbox"/>
mime-document-id		<input type="checkbox"/>

The following example shows the corresponding BPML parameters for the MIME service encoding GPM parameters.

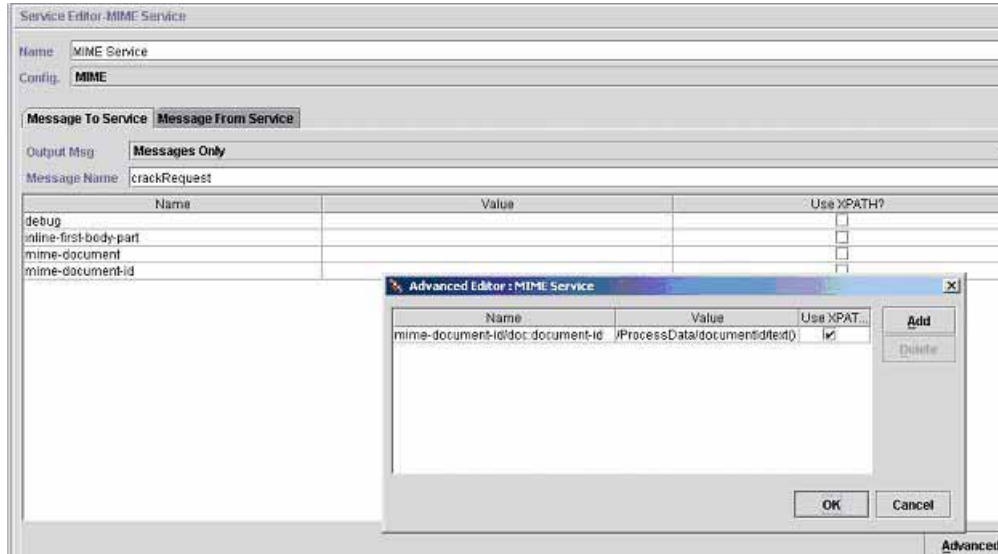
```
<process name="ExampleMimeEncodeGPM">
  <sequence>
    <operation name="MIME encode documents">
      <participant name="MIME"/>
      <output message="packRequest">
      <assign to="mime-document" from="nodeInProcessData/mime:message"/>
    
```

```

</output>
<input message="packResponse">
<assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

The following screen shows an example of a graphical view of the GPM parameters for decoding a document using the MIME service. You must set the mime-document-id/doc:document-id parameter for the document being decoded using the Advanced Editor.



The following example shows the corresponding BPML parameters for the MIME service decoding GPM parameters.

```

<process name="ExampleMimeDecodeGPM">
  <sequence>
    <operation name="MIME Service">
      <participant name="MIME"/>
      <output message="crackRequest">
        <assign to="mime-document-id/doc:document-id"
from="/ProcessData/documentId/text()" />
      </output>
      <input message="crackResponse">
        <assign to="decodedDocument" from="mime-document/*" />
      </input>
    </operation>
  </sequence>
</process>

```

The following table describes the fields used to configure the MIME service in the GPM. The Message Name in the configuration can be set either to packRequest or crackRequest:

Set the *Message Name* (or *output message*) of the MIME service to packRequest when you want to MIME-encode documents.

Set the *Message Name* (or *output message*) of the MIME service to `crackRequest` when you want to MIME-decode documents.

where *Message Name* is the GPM parameter and *output message* is the BPML parameter.

Field	Description
Config	Name of the service configuration.
debug	Indicates if debug information needs to be printed. Valid values are true and null.
inline-first-body-part	Whether the first body part is to be inlined. Valid values are true and null. Required when service starts with output message set to <code>crackRequest</code> . Note: <code>inline-first-body-part = true</code> implies that the first body part of the incoming MIME multipart document is XML and will be available in the process data upon the completion of this service.
mime-document	Node that needs to be MIME encoded. Required when service starts with output message set to <code>packRequest</code> .
mime-document-id	Node containing the document-id referencing the multipart document. Required when service starts with output message set to <code>crackRequest</code> .
addFilenameEscapes	Indicates if the filename parameter in the Content-Disposition field has its backslashes "unescaped". Preserves the backslashes in the filename. Optional.

Troubleshooting MIME Service Configurations

Consider these points when troubleshooting MIME service configurations:

The construction of the MIME multipart document is dependent on the structure of the XML representation of the MIME message in process data that is being passed to the service. If the service fails, it is often because the XML representation of the MIME message is invalid. See *Setting up Process Data for MIME Encoding* on page 969 for information about setting up the XML representation.

You must keep track of the document Ids of all the payloads you want to encode. You need to specify these document Ids in the XML representation of the MIME message.

To enable base64 encoding of attachments, the Content-Transfer-Encoding of the attachment needs to be specified in the header. To do this, add the following line to the `<mime:header>` tag of the attachment:

```
<mime:header name="Content-Transfer-Encoding">base64</mime:header>
```

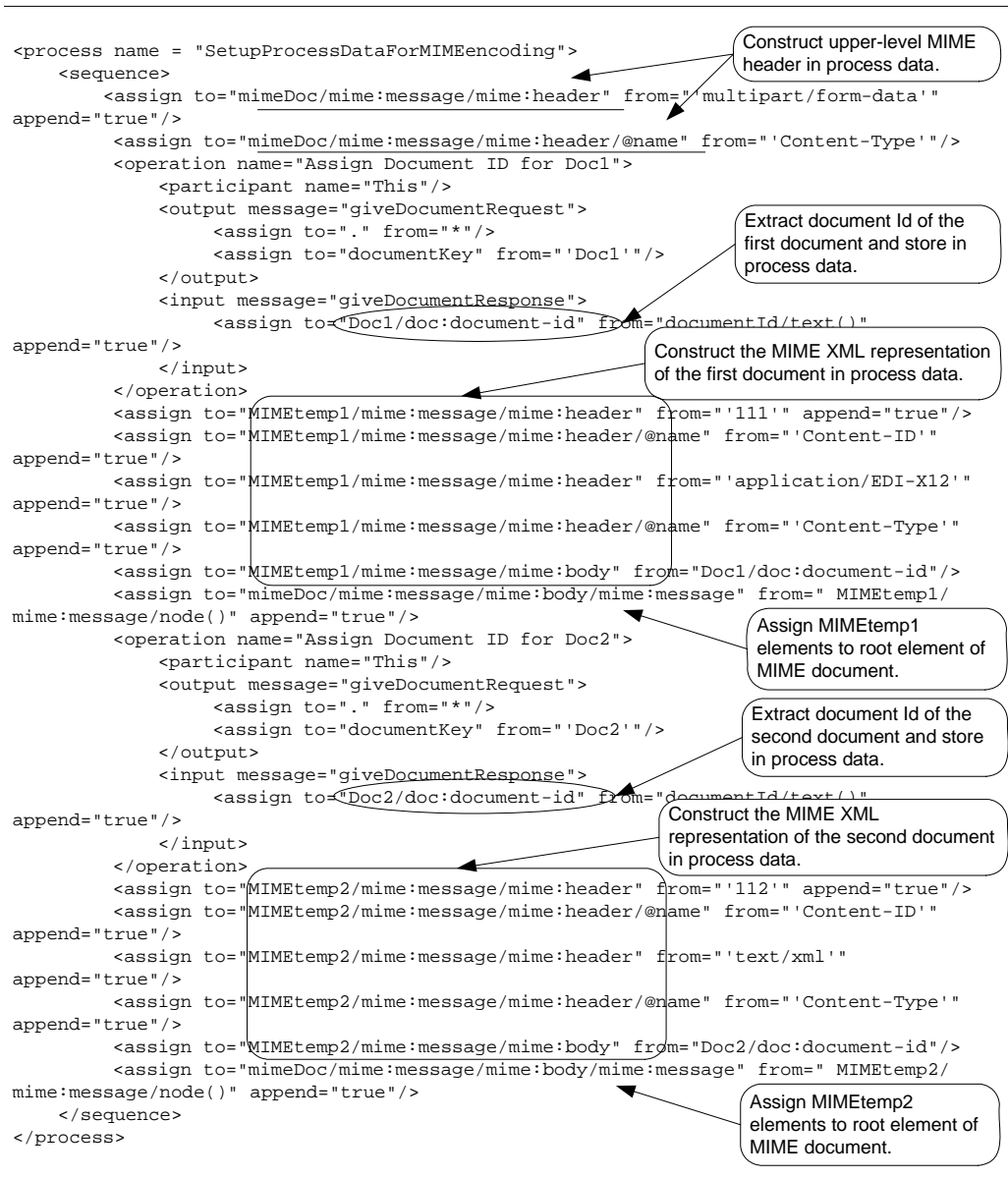
Usage Examples

This section contains additional examples using the MIME service, including information about creating the XML representation of the MIME message.

Setting up Process Data for MIME Encoding

The following example illustrates how you can create the XML representation of the MIME message in process data for input to the MIME service for encoding documents. This business process assumes that you

have two documents you want to MIME encode, one Microsoft Word document and one XML document. It is assumed that these files were input to the business process and stored in process data as Doc1 and Doc2 (not shown).



Running the above business process creates the following information in process data, which is input to the MIME service:

```

<mimeDoc>
  <mime:message xmlns:mime="http://www.company.com/company-name/mime/v0.5">
    <mime:header name="Content-Type">multipart/form-data</mime:header>
    <mime:body>
      <mime:message>
        <mime:header name="Content-ID">111</mime:header>
        <mime:header name="Content-Type">application/EDI-X12</mime:header>
        <mime:body>
          <doc:document-id xmlns:doc="http://www.company.com/company-name/
document-id">servername:5f6303:fb326b59ac:-6f71</doc:document-id>
        </mime:body>
      </mime:message>
      <mime:message>
        <mime:header name="Content-ID">112</mime:header>
        <mime:header name="Content-Type">text/xml</mime:header>
        <mime:body>
          <doc:document-id xmlns:doc="http://www.company.com/company-name/
document-id">servername:5f6303:fb326b59ac:-6f5e</doc:document-id>
        </mime:body>
      </mime:message>
    </mime:body>
  </mime:message>
</mimeDoc>

```

Represents MIME multipart for all documents included in the MIME document

Represents MIME multipart for MS Word document

Represents MIME multipart for text document

MIME Decoding

The following example illustrates a MIME document that needs decoding. The following MIME document is input to the business process.

```

Content-Type: multipart/form-data; boundary="serverName::5f6303:fb4edbb774:-5b83::multipart.boundary.string"
Content-Length: 19757

--serverName::5f6303:fb4edbb774:-5b83::multipart.boundary.string
Content-ID: 111
Content-Type: application/EDI-X12
_Microsoft Word Document
MSWordDoc_Word.Document.89q
--serverName::5f6303:fb4edbb774:-5b83::multipart.boundary.string
Content-ID: 112
Content-Type: text/xml

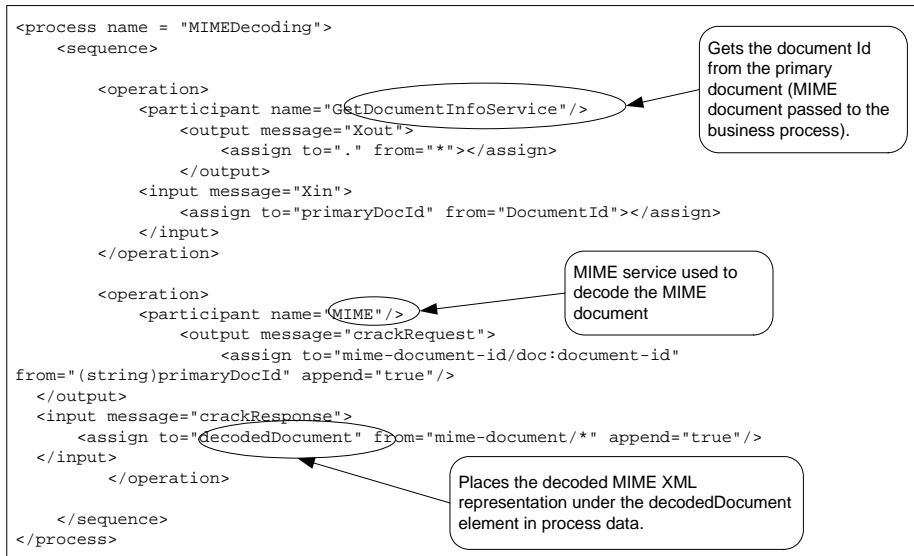
<Text>
</Text>
--serverName::5f6303:fb4edbb774:-5b83::multipart.boundary.string--

```

Multipart for MS Word document

Multipart for text document

The following example illustrates a business process in BPML format that uses the MIME service to decode the MIME document.



The following example contains an XML representation of the decoded MIME document in process data after the business process has run.

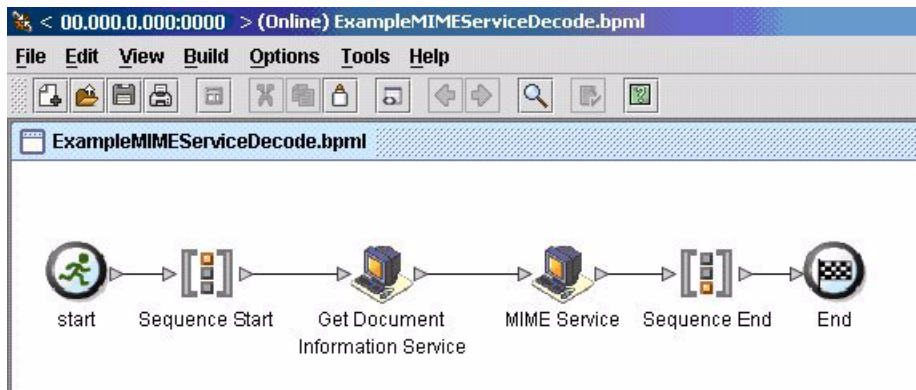
```

<decodedDocument>
  <mime:message xmlns:mime="http://www.company.com/company-name/mime/v0.5">
    <mime:header
      name="content-type">multipart/form-data<mime:parameter
name="boundary">servername:5f6303:fb4edbb774:-5a60::multipart.boundary.string</mime:parameter>
    </mime:header>
    <mime:header name="content-length">19757</mime:header>
    <mime:body>
      <mime:message>
        <mime:header name="content-id">111</mime:header>
        <mime:header name="content-type">application/edi-x12</mime:header>
        <mime:body>
          <doc:document-id
xmlns:doc="http://www.company.com/company-name/document-id">servername:5f6303:fb4edbb774:-5a42</doc:document-id>
        </mime:body>
      </mime:message>
      <mime:message>
        <mime:header name="content-id">112</mime:header>
        <mime:header name="content-type">text/xml</mime:header>
        <mime:body>
          <doc:document-id
xmlns:doc="http://www.company.com/company-name/document-id">servername:5f6303:fb4edb774:-5a40</doc:document-id>
        </mime:body>
      </mime:message>
    </mime:body>
  </mime:message>

```

```
</mime:message>  
</decodedDocument>
```

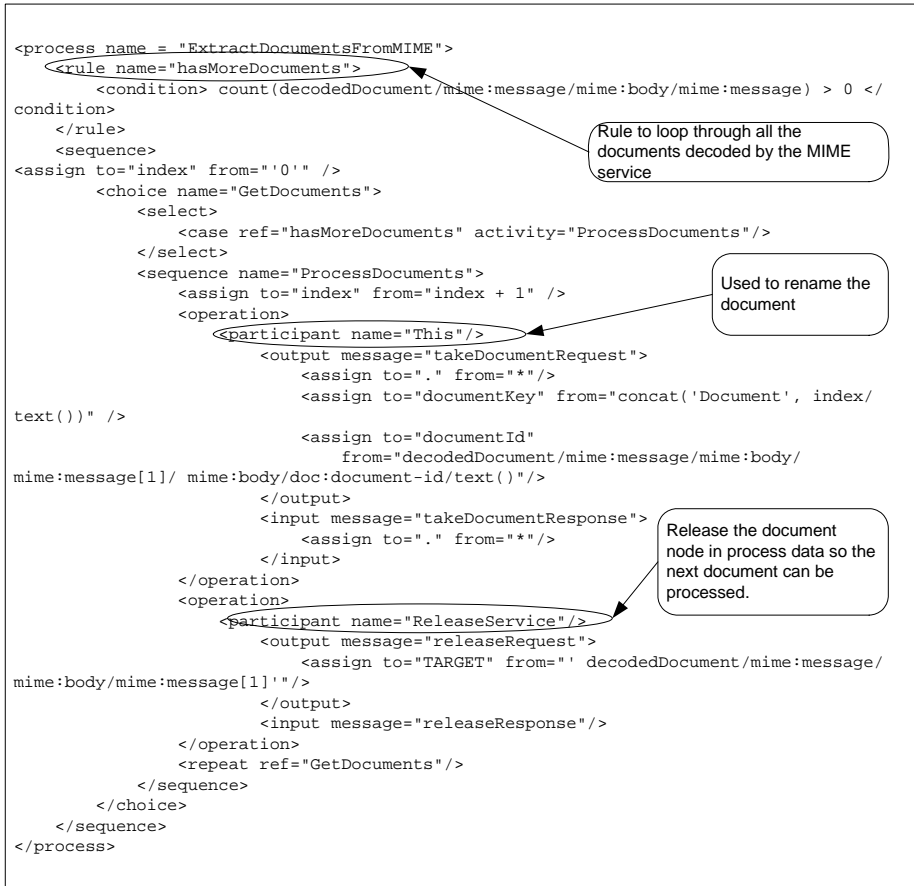
The following example illustrates the business process in the GPM:



Extracting the MIME Service Decoded Documents

The following example illustrates how you can extract the documents that have been decoded by the MIME Service. This business process loops through all the documents that have been decoded by the MIME

service and renames them to Document1 and Document2. The XML representation of the decoded MIME document in process data is input to this business process.



MQRFH2 Service

The MQRFH2 service passes JMS header information through WebSphereMQ.

System Name	MQRFH2
Graphical Process Modeler (GPM) categories)	Messaging, All
Description	Adds the MQ Rules and Formatting Header 2 to the beginning of the primary document.
Business usage	Passes JMS header information through WebSphereMQ.
Usage example	Invoke this service before calling the WebSphereMQ Adapter or WebSphereMQ Suite.
Preconfigured?	No
Requires third party files?	Requires the jars that are required by the WebSphereMQ Adapter/Suite (com.ibm.mq.jar).
Platform availability	All supported Application platforms
Related services	WebSphereMQ Adapter or WebSphereMQ Suite.
Application requirements	None
Initiates business processes?	This service does not perform any bootstrapping.
Invocation	Create an instance of this service and invoke it in a business process.

Business process context considerations	<p>This service looks for the following information in process data in order to build the MQRFH2:</p> <pre data-bbox="485 270 854 573"> <ProcessData> <RFH2> <mcd>....</mcd> <jms>....</jms> <usr>....</usr> <mqext>....</mqext> </RFH2> </ProcessData> </pre> <p>All of the above folders (mcd, jms, usr, mqext) are optional. Only specify the ones you want to use. If no folders are found, the following is used:</p> <pre data-bbox="485 659 902 684"> <mcd><Msd>none</Msd></mcd> </pre> <p>This link contains information regarding the various folders: http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp?topic=/com.ibm.etools.mft.doc/aq06920_.htm</p> <p>For the WebSphereMQ Server to properly handle messages with MQRFH2, the MQMD.format field must be set to MQFMT_RF_HEADER_2 (MQHRF2). Use the following parameter to set this for:</p> <ul style="list-style-type: none"> ◆ WebSphereMQ Adapter: <pre data-bbox="524 957 1289 982"> <assign to="snd_MQMD_format" from="'MQHRF2 '"/> </pre> ◆ WebSphereMQ Suite (PutMessage): <pre data-bbox="485 1041 1268 1066"> <assign to="wsmq_MQMD_format" from="'MQHRF2 '"/> </pre>
Returned status values	<ul style="list-style-type: none"> ◆ Success: The service completed successfully. ◆ Error: The service failed with an error.
Restrictions	None
Testing considerations	No special considerations for testing/debugging. This service does not currently log any debug messages.

Implementing the MQRFH2 Service

To implement the MQRFH2 service, complete the following tasks:

1. Create a configuration of the MQRFH2 service. See *Managing Services and Adapters*.
2. Specify field settings for the service configuration in the GPM as necessary.
3. Use the service in a business process.

Configuring the MQRFH2 Service

You must create a configuration of the service in the Admin Console, then specify field settings for the new configuration of the service in the GPM. The following table contains the parameters available for the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
mqrh2_msgFormat	Format name of the message payload data. Valid value is Any valid format type (see MQ manual). Default value is MQFMT_STRING. Optional.
mqrh2_buffersize	Buffer size used for reading the PrimaryDocument. Default is 5120. Optional.

Business Process Example

The following example illustrates how the MQRFH2 service can be used in a business process:

```
<process name="MQRFH2test">
  <sequence>
    <assign to="RFH2/mcd" from="<mcd><Msd>none</Msd></mcd>" />
    <operation name="MQRFH2 test">
      <participant name="MQRFH2" />
      <output message="toService"><assign to="." from="*" /></output>
      <input message="fromService"><assign to="." from="*" /></input>
    </operation>
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession" />
      <output message="toService">
        <assign to="wsmq_hostname">someMQhost</assign>
        <assign to="wsmq_channel">someChannel</assign>
      </output>
      <input message="fromService"><assign to="." from="*" /></input>
    </operation>
    <sequence name="SessionOpened">
      <operation name="Open Queue">
        <participant name="WSMQ_OpenQueue" />
        <output message="toService">
          <assign to="." from="*" />
          <assign to="wsmq_qname">queueName</assign>
          <assign to="wsmq_MQOO_type">PUT</assign>
          <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
        </output>
        <input message="fromService" />
      </operation>
      <operation name="Put Message">
        <participant name="WSMQ_PutMessage" />
        <output message="toService">
          <assign to="." from="*" />
          <assign to="wsmq_qname">queueName</assign>
          <assign to="wsmq_MQMD_format" from="'MQRHF2  '"/>
        </output>
        <input message="fromService"><assign to="." from="*" /></input>
      </operation>
    </sequence>
  </sequence>
</process>
```

```

</operation>
<operation name="Commit the PUT">
  <participant name="WSMQ_Commit"/>
  <output message="toService"><assign to="." from="*" /></output>
  <input message="fromService"><assign to="." from="*" /></input>
</operation>
<sequence name="CloseSession">
  <operation name="Close Session">
    <participant name="WSMQ_CloseSession"/>
    <output message="toService"><assign to="." from="*" /></output>
    <input message="fromService"/>
  </operation>
</sequence> <!-- end of CloseSession sequence -->
<onFault>
  <operation name="Make sure session is closed">
    <participant name="WSMQ_CloseSession"/>
    <output message="toService"><assign to="." from="*" /></output>
    <input message="fromService"/>
  </operation>
</onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence>
</process>

```

MSMQ Adapter (Build 4300 - Build 4317)

The following table provides an overview of the MSMQ adapter:

System name	MSMQ
Graphical Process Modeler (GPM) categories	All Services, Custom, Sync Mode, Transactional Mode Note: This adapter will not display in the GPM stencils until you create a configuration of the adapter in the Application Admin UI.
Description	<p>Sends messages to and reads messages from a remote Microsoft® Message Queue. You can configure the MSMQ adapter to send and retrieve messages:</p> <ul style="list-style-type: none">◆ Send mode – Sends a message from a business process to a named queue on a specified Microsoft Windows® MSMQ server. The MSMQ adapter also returns pertinent message metadata to the business process after sending the message.◆ Retrieve mode – In a business process, the MSMQ adapter retrieves messages from a specified queue on a specified MSMQ server. When in retrieve mode, the MSMQ adapter continues to retrieve messages until the queue is empty. The MSMQ adapter also runs a specified business process for each message in the queue and passes the message body and selected metadata parameters to the business process that is running.
Business Usage	Processing messages from an MSMQ server as part of a business process.
Preconfigured?	No
Requires third party files?	Yes. Requires Jacob.lib (Com-java bridge package). This is included in the msmqbundle_4_1_0.jar, which is included with Application. See <i>Implementing the MSMQ Adapter</i> on page 981.
Platform availability	All supported Application platforms
Related services	No
Application Requirements	<ul style="list-style-type: none">◆ The MSMQ adapter supports MSMQ version 1.1 and later. The MSMQ adapter supports 40-bit encryption for versions 1.1 and later and 128-bit encryption in versions 2.0 and later.
Initiates business processes?	Yes, when using the RETRIEVE action (reading a message from the queue).
Invocation	Runs as a service within a business process.
Returned status values	Success – The run was successful for the specified action. Error – An error occurred when the specified action was executed.

Restrictions	<ul style="list-style-type: none"> ◆ The MSMQ adapter supports single message transactions for guaranteed exactly-once delivery of messages. Other transaction modes are not supported. ◆ The MSMQ adapter supports external transactions to MSMQ; that is, a transaction from Application for a SEND action. This only applies to transactional queues in MSMQ. ◆ The MSMQ adapter does not support MSMQ acknowledgements.
Persistence Level	<p>Full (default).</p> <p>Note: When running in transactional mode, this setting is overridden at the business process level and set to Zero persistence.</p>

Requirements

To use the MSMQ adapter, you need:

The connection from Application to the system where the MSMQ server resides.

The port number of the system where the MSMQ server resides. This port number is used by the MSMQ adapter.

The queue must be preconfigured in the MSMQ server. This queue name is used by the MSMQ adapter. The queue can be transactional or non-transactional, depending on how you need to use the MSMQ adapter.

How the MSMQ Adapter Works

The following steps summarize how the MSMQ adapter works in a business process within Application:

1. The MSMQ adapter extracts the message body from the incoming business process context and sends it to a remote system Microsoft Message Queue.
2. The MSMQ adapter creates a correlation ID and returns it in the process data elements.
3. The adapter retrieves any response from the Microsoft Message Queue and places it back into the business process context.
4. The adapter passes the updated business process context back to the business process and places any MSMQ elements into process data.

The following table contains the MSMQ elements placed into the process data after the MSMQ adapter runs:

MSMQ Element	Description
MSMQ_ARRIVEDTIME	When the message arrived at the queue.
MSMQ_BODY_TYPE	Whether the message is a string or byte array.
MSMQ_CORRELATIONID	ID number used to correlate related messages.
MSMQ_EXTENSION	A place to put additional information associated with the message.
MSMQ_FIRST_IN_XACT	Whether the message was the first message in a transaction.

MSMQ_LAST_IN_XACT	Whether the message was the last message in a transaction.
MSMQ_XACTID	Message ID from a transaction queue.
MSMQ_LABEL	Label for the message.
MSMQ_MSGID	Message ID from a regular queue.
MSMQ_PRIORITY	Message priority level.
MSMQ_RETURNED_MESSAGE	Retrieved message.
MSMQ_XACTIONAL_QUEUE	Name of the transactional queue from which the message was received.

Implementing the MSMQ Adapter

To implement the MSMQ adapter, complete the following tasks:

1. Activate your license for the MSMQ adapter. See *An Overview of Implementing Services*.
2. Install the MSMQPrime component.
3. Create and configure an MSMQ adapter configuration. For general instructions on creating a service configuration, see *Creating a Service Configuration*. For descriptions of the MSMQ adapter configuration parameters, see *Configuring the MSMQ Adapter* on page 982.
4. Use the MSMQ adapter in a business process.

Installing MSMQPrime

A component of the MSMQ adapter, MSMQPrime, is deployed on the host where the MSMQ server resides. The MSMQ adapter connects to the MSMQPrime component, which in turn, performs the send and receive actions and interacts with the MSMQ server. MSMQPrime listens on the port that is part of the MSMQ adapter configuration.

To install MSMQPrime, complete the following steps:

1. Locate `msmqbundle_4_1_0.jar` under `install_dir/installed_data/patch/files/client/msmq`.
2. On the Windows MSMQ server host, create a folder for MSMQPrime. For example, `C:\MSMQ`.
3. Copy `msmqbundle_4_1_0.jar` to the folder you just created.
4. Change directory to that folder, and use `winzip` to unbundle the `.jar` file.
5. Install the Java `jdk` version `1.4.2_06`.
6. Modify `start_msmqPrime.cmd` to use the folder you created in step 2.
7. Have `java` point to the installed `jdk` directory.
8. Change `MSMQ_SERVER_PORT` in `msmqprime.properties`, if necessary. The default is `8085`. This is the port `msmqPrime` will run on.
9. Run `start_msmqPrime.cmd`. This script should be run by the user who has permission to create queues, read, and send messages to the MSMQ server. This process must be running continually if your MSMQ adapter needs to access it. It is recommended to convert it to an automatically started Windows service.

10. Create a configuration of the MSMQ adapter in your application and configure it to point to this msmqPrime.
11. Configure a service instance of MSMQAdapter in the application to point to this msmqPrime.
12. Verify that the MSMQ adapter configuration is talking to this msmqPrime by including it in a business process and running it.
13. If desired for testing purposes, turn on debug mode in msmqPrime with this command:

```
start_msmqPrime.cmd debugon
```

The debugon option generates detailed logs.

14. To start msmqPrime with the debug mode turned on, use this command:

```
start_msmqPrime.cmd
```

Note: The msmqbundle_4_1_0.jar you use to create the MSMQPrime component must be from the same Application installation as the MSMQ adapter that it will “talk” to.

Configuring the MSMQ Adapter

To configure the MSMQ adapter, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the MSMQ adapter in Application:

Note: This table contains configuration parameters for both Send and Retrieve. The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Hostname (HostName)	Host name of the MSMQ server with which this configuration of the adapter communicates. Send and Retrieve parameters. Required.
PortNumber (PortNumber)	The port number of the MSMQ server host. Required.
Queue Path Name (QueuePathName)	Typical queue path name in the form <i>machineName\queueName</i> . Send and Retrieve parameters. Required.

Field	Description
EX_TRANSACTION	To include send action in the Application engine transaction. Valid values are TRUE and FALSE (default). Send parameter. Optional.

GPM Configuration

The following table describes the fields used to configure both Send and Retrieve configurations for the MSMQ adapter in the GPM:

Field	Description
Config	Name of the adapter configuration.
Action	Values are Send and Retrieve. Required if the action is Retrieve. If the action is Retrieve, specify business process. If the Action is Send, specify body type.
BodyType	Valid values are String (default) and Byte array. Send parameter. Optional.
BusinessProcessName	Business process that is started upon receipt of a message from the queue. Retrieve parameter. Required.
Delivery	Valid values are Recoverable (default) and Express. Send parameter. Optional.
EncryptionAlgorithm	CALG_RC2 (default) or CALG_RC4. Send parameter. Optional.
MaxTimeToReachQueue	Maximum time allowed for a message to reach its destination. Send parameter. Default is 300 seconds. Optional.
MaxTimeToReceive	Maximum time, in seconds, for a message to be received before it is discarded from the queue. Send parameter. Default is 0. Optional. Note: If set to 0, the message is not discarded.
MessageLabel	Label for message being sent to queue. Optional.
MessagePriority	Valid values are 0 - 7, where 7 represents the highest priority messages in the queue. The highest priority messages are received first. Send parameter. Default is 3. Optional. Note: This parameter setting does not apply to messages sent to a transactional queue, which will always have a message priority of 0.
PrivacyLevel	Used to request encryption. Valid values are None (default), Base (40-bit) and Enhanced (128-bit, supported in MSMQ version 2.0 and later). Send parameter. Optional.

Business Process Examples

The following examples illustrate using the MSMQ adapter for send and receive actions:

Send Action

```
<process name = "MSMQ_Base">
  <sequence name="Test Sequence">
    <operation name="MSMQ Commn">
```

```

<participant name="MSMQAdapter"/>
<output message="outmsg">
  <assign to="Action">SEND</assign>
  <assign to="PortNumber">0000</assign>
  <assign to="QueuePathName">server1\testqueue</assign>
  <assign to="MaxTimeToReceive">3600</assign>
  <assign to="MaxTimeToReachQueue">3600</assign>
  <assign to="MessagePriority">3</assign>
  <assign to="Delivery">RECOVERABLE</assign>
  <assign to="PrivacyLevel">Base</assign>
  <assign to="MessageLabel">Base</assign>
  <assign to="BodyType">BYTE ARRAY</assign>
  <assign to="GIS_TRANSACTION">TRUE</assign>

  <assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
</sequence>
</process>

```

Retrieve Action

```

<process name = "MSMQ_Receive">
  <sequence name="Test Sequence">
    <operation name="MSMQ Commn">
      <participant name="MSMQAdapter"/>
      <output message="outmsg">
        <assign to="Action" from="'RETRIEVE'" />
        <assign to="PortNumber" from="'0000'" />
        <assign to="QueuePathName" from="'server1\testqueue'" />
        <assign to="MessageLabel" from="'Receive'" />
        <assign to="BusinessProcessName" from="'MSMQ_FileSystem'" />
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>

```

MSMQ Adapter (Build 4318 or higher)

The following table provides an overview of the MSMQ adapter:

System name	MSMQ
Graphical Process Modeler (GPM) categories	All Services, Custom, Sync Mode, Transactional Mode Note: This adapter will not display in the GPM stencils until you create a configuration of the adapter in the Application Admin UI.
Description	<p>Sends messages to and reads messages from a remote Microsoft® Message Queue. You can configure the MSMQ adapter to send and retrieve messages:</p> <ul style="list-style-type: none">◆ Send mode – Sends a message from a business process to a named queue on a specified Microsoft Windows® MSMQ server. The MSMQ adapter also returns pertinent message metadata to the business process after sending the message.◆ Retrieve mode – In a business process, the MSMQ adapter retrieves messages from a specified queue on a specified MSMQ server. When in retrieve mode, the MSMQ adapter continues to retrieve messages until the queue is empty. The MSMQ adapter also runs a specified business process for each message in the queue and passes the message body and selected metadata parameters to the business process that is running.
Business Usage	Processing messages from an MSMQ server as part of a business process.
Preconfigured?	No
Requires third party files?	Yes. Requires Jacob.lib (Com-java bridge package). This is included in the msmqbundle_4_1_0.jar, which is included with Application. See <i>Implementing the MSMQ Adapter</i> on page 987.
Platform availability	All supported Application platforms
Related services	No
Application Requirements	<ul style="list-style-type: none">◆ The MSMQ adapter supports MSMQ version 1.1 and later. The MSMQ adapter supports 40-bit encryption for versions 1.1 and later and 128-bit encryption in versions 2.0 and later.
Initiates business processes?	Yes, when using the RETRIEVE action (reading a message from the queue).
Invocation	Runs as a service within a business process.
Returned status values	Success – The run was successful for the specified action. Error – An error occurred when the specified action was executed.

Restrictions	<ul style="list-style-type: none"> ◆ The MSMQ adapter supports single message transactions for guaranteed exactly-once delivery of messages. Other transaction modes are not supported. ◆ The MSMQ adapter supports external transactions to MSMQ; that is, a transaction from Application for a SEND action. This only applies to transactional queues in MSMQ. ◆ The MSMQ adapter does not support MSMQ acknowledgements.
Persistence Level	<p>Full (default).</p> <p>Note: When running in transactional mode, this setting is overridden at the business process level and set to Zero persistence.</p>

Requirements

To use the MSMQ adapter, you need:

The connection from Application to the system where the MSMQ server resides.

The port number of the system where the MSMQ server resides. This port number is used by the MSMQ adapter.

The queue must be preconfigured in the MSMQ server. This queue name is used by the MSMQ adapter. The queue can be transactional or non-transactional, depending on how you need to use the MSMQ adapter.

How the MSMQ Adapter Works

The following steps summarize how the MSMQ adapter works in a business process within Application:

1. The MSMQ adapter extracts the message body from the incoming business process context and sends it to a remote system Microsoft Message Queue.
2. The MSMQ adapter creates a correlation ID and returns it in the process data elements.
3. The adapter retrieves any response from the Microsoft Message Queue and places it back into the business process context.
4. The adapter passes the updated business process context back to the business process and places any MSMQ elements into process data.

The following table contains the MSMQ elements placed into the process data after the MSMQ adapter runs:

MSMQ Element	Description
MSMQ_ARRIVEDTIME	When the message arrived at the queue.
MSMQ_BODY_TYPE	Whether the message is a string or byte array.
MSMQ_CORRELATIONID	ID number used to correlate related messages.
MSMQ_EXTENSION	A place to put additional information associated with the message.
MSMQ_FIRST_IN_XACT	Whether the message was the first message in a transaction.

MSMQ_LAST_IN_XACT	Whether the message was the last message in a transaction.
MSMQ_XACTID	Message ID from a transaction queue.
MSMQ_LABEL	Label for the message.
MSMQ_MSGID	Message ID from a regular queue.
MSMQ_PRIORITY	Message priority level.
MSMQ_RETURNED_MESSAGE	Retrieved message.
MSMQ_XACTIONAL_QUEUE	Name of the transactional queue from which the message was received.

Implementing the MSMQ Adapter

To implement the MSMQ adapter, complete the following tasks:

1. Activate your license for the MSMQ adapter. See *An Overview of Implementing Services*.
2. Install the MSMQPrime component.
3. Create and configure an MSMQ adapter configuration. For general instructions on creating a service configuration, see *Creating a Service Configuration*. For descriptions of the MSMQ adapter configuration parameters, see *Configuring the MSMQ Adapter* on page 988.
4. Use the MSMQ adapter in a business process.

Installing MSMQPrime

MSMQPrime is a component of the MSMQ adapter and should be deployed within the same Microsoft network where the MSMQ server resides. It can be co-located with the MSMQ server, the Application server, or an independent server. It must be able to reach the MSMQ server queue utilizing the Microsoft resource naming convention. The MSMQ adapter connects to the MSMQPrime component, which in turn, performs the send and receive actions and interacts with the MSMQ server. MSMQPrime listens on the port that is part of the MSMQ adapter configuration.

To install MSMQPrime, complete the following steps:

1. Locate `msmqbundle_xxx.jar` in the Application under `<INSTALL_DIR>/client/msmq` folder.
2. On the Windows MSMQ server host, create a folder for MSMQPrime. For example, `C:\MSMQ`.
3. Copy `msmqbundle_xxx.jar` to the folder you just created.
4. Change directory to that folder, and use `winzip` to unbundle the `.jar` file.
5. Copy `Jacob.dll` from the `msmqbundle*.jar/Jacob/1_7` folder to the `C:\WINDOWS\system32` folder.
6. Install the Java jdk version 1.5.0_11. Note the installation path.
7. Modify `start_msmqPrime.cmd` to use the folder you created in step 2. Set the `MSMQADAPTER` parameter to the folder you created in step 2. Set the `JAVA` parameter to point to the bin directory in the Java path created in step 5; that is, `C:\jdk1.5.0_11\bin`.

Note: If Java is installed in the default installation folder in C:\Program Files\Java\jdk1.5.0_11, you have to reference it as C:\Progra~1\Java\jdk1.5.0_11\bin.

8. Change `MSMQ_SERVER_PORT` in `msmqprime.properties`, if necessary. The default is 8085. This is the port `msmqPrime` will run on.
9. Run `start_msmqPrime.cmd`. This script should be run by the user who has permission to create queues, read, and send messages to the MSMQ server. This process must be running continually if your MSMQ adapter needs to access it. It is recommended to convert it to an automatically started Windows service.
10. Create a configuration of the MSMQ adapter in your Application and configure it to point to this `msmqPrime`.
11. Configure a service instance of MSMQ adapter in the Application to point to this `msmqPrime`.
12. Verify that the MSMQ adapter configuration is talking to this `msmqPrime` by including it in a business process and running it.
13. If desired for testing purposes, turn on debug mode in `msmqPrime` with the following command by passing `-debug` as an argument to `MSMQPrimaImpl` in `start_msmq.cmd`.

The `debugon` option generates detailed logs.

Note: The `msmqbundle_xxx.jar` you use to create the `MSMQPrime` component must be from the same Application installation as the MSMQ adapter that it will talk to. You need to redeploy the `msmqbundle_xxx.jar` to the Windows MSMQ server host when an Application patch is installed. This ensures the new code changes are synchronized with `MSMQPrime`.

Configuring the MSMQ Adapter

To configure the MSMQ adapter, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the MSMQ adapter in Application:

Note: This table contains configuration parameters for both Send and Retrieve. The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.

Field	Description
Hostname (HostName)	Host name of the MSMQ server with which this configuration of the adapter communicates. Send and Retrieve parameters. Required.
PortNumber (PortNumber)	The port number of the MSMQ server host. Required.
Queue Path Name (QueuePathName)	Typical queue path name in the form <i>machineName\queueName</i> . Send and Retrieve parameters. Required.
EX_TRANSACTION	To include send action in the Application engine transaction. Valid values are TRUE and FALSE (default). Send parameter. Optional.

GPM Configuration

The following table describes the fields used to configure both Send and Retrieve configurations for the MSMQ adapter in the GPM:

Field	Description
Config	Name of the adapter configuration.
Action	Values are Send and Retrieve. Required if the action is Retrieve. If the action is Retrieve, specify business process. If the Action is Send, specify body type.
BodyType	Valid values are String (default) and Byte array. Send parameter. Optional.
BusinessProcessName	Business process that is started upon receipt of a message from the queue. Retrieve parameter. Required.
ProcessMode	The mode where the business process is started and executed. Valid values: <ul style="list-style-type: none"> ◆ AsyncBootstrap - Mode that provides default Sterling Integrator functionality. The business processes started by the adapter are placed on the Sterling Integrator queues and executed asynchronously to the adapter. ◆ FifoBootstrap - Mode that executes business processes in First-In, First-Out order. See <i>FIFO Message Processing Enhancement for Sterling Integrator 4.3</i> for additional information about this processing mode.
FIFORouteBP	Specify the name of the business process that will be executed to determine the FIFO routing key. Note: This option is only available when the adapter is in FIFO processing mode.
Delivery	Valid values are Recoverable (default) and Express. Send parameter. Optional.
EncryptionAlgorithm	CALG_RC2 (default) or CALG_RC4. Send parameter. Optional.
MaxTimeToReachQueue	Maximum time allowed for a message to reach its destination. Send parameter. Default is 300 seconds. Optional.

Field	Description
MaxTimeToReceive	Maximum time, in seconds, for a message to be received before it is discarded from the queue. Send parameter. Default is 0. Optional. Note: If set to 0, the message is not discarded.
MessageLabel	Label for message being sent to queue. Optional.
MessagePriority	Valid values are 0 - 7, where 7 represents the highest priority messages in the queue. The highest priority messages are received first. Send parameter. Default is 3. Optional. Note: This parameter setting does not apply to messages sent to a transactional queue, which will always have a message priority of 0.
PrivacyLevel	Used to request encryption. Valid values are None (default), Base (40-bit) and Enhanced (128-bit, supported in MSMQ version 2.0 and later). Send parameter. Optional.

Business Process Examples

The following examples illustrate using the MSMQ adapter for send and receive actions:

Send Action

```
<process name = "MSMQ_Base">
  <sequence name="Test Sequence">
    <operation name="MSMQ Commn">
      <participant name="MSMQAdapter" />
      <output message="outmsg">
        <assign to="Action">SEND</assign>
        <assign to="PortNumber">0000</assign>
        <assign to="QueuePathName">server1\testqueue</assign>
        <assign to="MaxTimeToReceive">3600</assign>
        <assign to="MaxTimeToReachQueue">3600</assign>
        <assign to="MessagePriority">3</assign>
        <assign to="Delivery">RECOVERABLE</assign>
        <assign to="PrivacyLevel">Base</assign>
        <assign to="MessageLabel">Base</assign>
        <assign to="BodyType">BYTE_ARRAY</assign>
        <assign to="GIS_TRANSACTION">TRUE</assign>

        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Retrieve Action

```
<process name = "MSMQ_Receive">
  <sequence name="Test Sequence">
    <operation name="MSMQ Commn">
```

```
<participant name="MSMQAdapter"/>
<output message="outmsg">
  <assign to="Action" from="'RETRIEVE'" />
  <assign to="PortNumber" from="'0000'" />
  <assign to="QueuePathName" from="'server1\testqueue'" />
  <assign to="MessageLabel" from="'Receive'" />
  <assign to="BusinessProcessName" from="'MSMQ_FileSystem'" />
  <assign to="." from="*" />
</output>
<input message="inmsg">
  <assign to="." from="*" />
</input>
</operation>
</sequence>
</process>
```

MSMQ Send Service

The following table provides an overview of the MSMQ Send service:

System name	MSMQ Send Service
Graphical Process Modeler (GPM) categories	All Services
Description	Sends messages to a remote Microsoft® Message Queue. Sends a message from a business process to a named queue on a specified Microsoft Windows® MSMQ server. The MSMQ Send service also returns pertinent message metadata to the business process after sending the message.
Business usage	Use the MSMQ Send Service to send a message to a remote Microsoft® Message Queue.
Preconfigured?	Yes
Requires third party files?	Yes. Requires <code>msmqbundle_xxx.jar</code> file. Application installation includes <code>msmqbundle_xxx.jar</code> file in the <code><INSTALL_DIR>/client/msmq</code> folder.
Platform availability	The MSMQ Send service is available for the following platforms <ul style="list-style-type: none">◆ Microsoft Windows◆ Sun Solaris◆ HP-UX◆ IBM-AIX◆ United Linux◆ RedHat AS◆ iSeries (OS/400)◆ zSeries (z/OS)
Related services	N/A
Application requirements	<ul style="list-style-type: none">◆ The connection from Application to the system where the MSMQ server resides.◆ The port number of the system where the MSMQ server resides. This port number is used by the MSMQ Send service.◆ The queue must be preconfigured in the MSMQ server. This queue name is used by the MSMQ Send service. The queue can be transactional or non-transactional, depending on how you need to use the MSMQ Send service.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Returned status values	Success - The run was successful for the specified action. Error - An error occurred when the specified action was executed.

Restrictions	<ul style="list-style-type: none"> ◆ The MSMQ Send service supports MSMQ version 1.1 and later. The MSMQ Send service supports 40-bit encryption for versions 1.1 and later and 128-bit encryption in versions 2.0 and later. ◆ The MSMQ Send service supports single message transactions for guaranteed exactly-once delivery of messages. Other transaction modes are not supported. ◆ The MSMQ Send service supports external transactions to MSMQ; that is, a transaction from Application for a SEND action. This only applies to transactional queues in MSMQ. ◆ The MSMQ Send service does not support MSMQ acknowledgements.
Persistence level	System Default (Full)

Implementing the MSMQ Send Service

To implement the MSMQ Send service, complete the following tasks:

1. Activate your license for the MSMQ Send service. For more information, see *An Overview of Implementing Services*.
2. Install the MSMQPrime component. For information on installing MSMQPrime, see *Installing MSMQPrime* on page 993.
3. Create and configure an MSMQ Send service configuration. For general instructions on creating a service configuration, see *Creating a Service Configuration*. For descriptions of the MSMQ Send service configuration parameters, see *Configuring the MSMQ Send Service* on page 995.
4. Use the MSMQ Send service in a business process.

Installing MSMQPrime

A component of the MSMQ Send service, MSMQPrime, is deployed on the host where the MSMQ server resides. The MSMQ Send service connects to the MSMQPrime component, which in turn, sends messages and interacts with the MSMQ server. MSMQPrime listens on the port that is part of the MSMQ Send service configuration.

To install MSMQPrime, complete the following tasks:

1. Locate `msmqbundle_xxx.jar` in the Application under `<INSTALL_DIR>/client/msmq` folder.
2. On the Windows MSMQ server host, create a folder for MSMQPrime. For example, `C:\MSMQ`.
3. Copy `msmqbundle_xxx.jar` to the folder you just created.
4. Change directory to that folder, and use winzip to unbundle the `.jar` file.
5. Install the Java jdk version 1.5.0_11. Note the installation path.

6. Modify `start_msmqPrime.cmd` to use the folder you created in step 2. Set the `MSMQADAPTER` parameter to the folder you created in step 2. Set the `JAVA` parameter to point to the bin directory in the Java path created in step 5; that is, `C:\jdk1.5.0_11\bin`.

Note: If Java is installed in the default installation folder in `C:\Program Files\Java\jdk1.5.0_11`, you have to reference it as `C:\Progra~1\Java\jdk1.5.0_11\bin`.

7. Change `MSMQ_SERVER_PORT` in `msmqprime.properties`, if necessary. The default is 8085. This is the port `msmqPrime` will run on.
8. Run `start_msmqPrime.cmd`. This script should be run by the user who has permission to create queues, read, and send messages to the MSMQ server. This process must be running continually if your MSMQ Send service needs to access it. It is recommended to convert it to an automatically started Windows service.
9. Create a configuration of the MSMQ Send service in your Application and configure it to point to this `msmqPrime`.
10. Configure a service instance of MSMQ Send service in the Application to point to this `msmqPrime`.
11. Verify that the MSMQ Send service configuration is talking to this `msmqPrime` by including it in a business process and running it.
12. If desired for testing purposes, turn on debug mode in `msmqPrime` with the following command by passing `-debug` as an argument to `MSMQPrimaImpl` in `start_msmq.cmd`.

The `debugon` option generates detailed logs.

Note: The `msmqbundle_xxx.jar` you use to create the `MSMQPrime` component must be from the same Application installation as the MSMQ Send service that it will talk to. You need to redeploy the `msmqbundle_xxx.jar` to the Windows MSMQ server host when an Application patch is installed. This ensures the new code changes are synchronized with `MSMQPrime`.

Installing MSMQPrime as a Windows Service

To install `MSMQPrime` as a Windows Service, complete the following tasks:

1. Unbundle the `msmqbundle_xxx.jar` file. A directory called `installJavaService` is created.
2. Modify the `installJavaService\InstallJavaService.cmd` script to set the `JAVA` parameter to the bin directory in the Java path created in step 6 of installing the `MSMQPrime` component. Set the `MSMQINSTALLDIR` parameter to the folder created in step 2 of installing the `MSMQPrime` component.
3. If you want to turn on the debug mode in `msmqPrime`, set the `PARAMS` to `debug`. By default, `debug` is turned off.
4. Run `installJavaService.cmd` to install `MSMQPrime` as a Windows service.
5. To start the service, from the Windows Services applet, select the New MSMQ Send service and select `Start`.
6. To stop the service, from the Windows Services applet, select the New MSMQ Send service and select `Stop`.

Uninstalling MSMQPrime as a Windows Service

To uninstall MSMQPrime as a Windows Service, run the following command:

```
installJavaService\uninstallJavaService.cmd
```

Configuring the MSMQ Send Service

To configure the MSMQ Send service, you must specify settings for the following fields:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the following options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .
Remote Host Name	Host name of the MSMQ server with which this configuration of the server communicates and where the MSMQPrime is installed. Required.
Port Number Of Remote Host	The port number of the MSMQPrime host. Required. Default value is 8085.
Queue Path Name	Typical queue path name configured in the MSMQ server in the form <code>Hostname\queueName</code> . Optional.
Document Storage Type	Indicate whether the message should be stored on the file system or in the database. Select one of the following options: <ul style="list-style-type: none">◆ System Default◆ Database◆ File System Required. Default value is System Default.

GPM Configuration

The following table describes the parameters you need to specify for configuring MSMQ Send service in the GPM:

Field	Description
Action	Specifies the type of action. Valid value is Send. Default value is Send. Required.
BodyType	Valid values are Byte array and String (default). Optional.

Field	Description
Delivery	Valid values are Express and Recoverable (default). Optional.
EncryptionAlgorithm	Valid values are CALG_RC2 and CALG_RC4. Optional.
EX_TRANSACTION	Indicates whether to include send action in the Application engine transaction. Valid values are True and False. Default value is False. Required.
MaxTimeToReachQueue	Maximum time allowed for a message to reach its destination. Default is 300 seconds. Optional.
MessageLabel	Label for message being sent to queue. Optional.
MessagePriority	Valid values are 0 - 7, where 7 represents the highest priority messages in the queue. The highest priority messages are received first. Default is 3. Optional. Note: This parameter setting does not apply to messages sent to a transactional queue, which will always have a message priority of 0.
PrivacyLevel	Used to request encryption. Valid values are None (default), Base (40-bit) and Enhanced (128-bit, supported in MSMQ version 2.0 and later). Optional.

Output from Service to Business Process

The following table contains the parameter passed from the MSMQ Send service to the business process:

Field	Description
MSMQ_CORRELATIO NID	ID number used to correlate related messages. Optional.

Business Process Example

The following example business process illustrates using the MSMQ Send service:

```
<process name = "MSMQ_Base">
  <sequence name="Test Sequence">
    <operation name="MSMQ Commn">
      <participant name="MSMQSend" />
      <output message="outmsg">
        <assign to="Action">SEND</assign>
        <assign to="PortNumber">8085</assign>
        <assign to="QueuePathName">lando.labsci.local\testqueuetx</assign>
        <assign to="MaxTimeToReachQueue">3600</assign>
        <assign to="MessagePriority">3</assign>
        <assign to="Delivery">RECOVERABLE</assign>
        <assign to="PrivacyLevel">Base</assign>
        <assign to="MessageLabel">Base</assign>
        <assign to="BodyType">BYTE ARRAY</assign>
        <assign to="GIS_TRANSACTION">TRUE</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

Network Report Reconciliation Service

The following table provides an overview of the Network Report Reconciliation service:

System name	NetworkReportReconciliation
Graphical Process Modeler (GPM) categories	All Services, EDI X12
Description	<p>Adds status correlation database records for EDI interchanges sent to the network and updates the status correlations reconciled from the network reports. Because this is a separate service, it can be used independently of the transport used to communicate with the network.</p> <p>Note: Currently only the Sterling Commerce Network is supported.</p>
Business usage	Enables a user to track EDI interchanges and their network status.
Usage example	You must use this service in a business process after EDI data is sent to the network for the initial status (Sent to network) correlation records to be added to Application. You must also use this service after the step that receives the network reports so that the service can reconcile them against the correlation records. You can use the Application Correlation Search function to view the network reconciliation status for EDI interchanges.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	No
Initiates business processes?	No
Invocation	Runs after sending EDI data to the network and after receiving reports from the network.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success–The service completed successfully.◆ Error–The service experienced a fatal error while processing.
Restrictions	<p>When this service runs after sending data to the network, the primary document identifier should not have changed since being enveloped because the correlation records are based on the same identifier created by the enveloping service.</p> <p>Network reports must be in data format, not human-readable format. They can either be in original or extended format but each record must be delimited by a carriage return, a linefeed, or a carriage return followed immediately by a linefeed.</p>
Persistence level	System default

Testing considerations	<p>1) A business process that generates EDI data, sends it to the network, and calls this service to create the initial correlation records.</p> <p>2) A business process that receives network reports and calls this service to process the reports and update the correlation records to reflect the network status.</p>
------------------------	---

How the Network Report Reconciliation Service Works

As part of the EDI enveloping process in Application, correlation records are automatically created and added to the correlation table, including InterchangeSenderID, InterchangeReceiverID, and InterchangeControlNumber.

When used in a business process that sends EDI interchanges to a network, the Network Report Reconciliation service collects additional correlation records to show that the data was sent to the network. These records are:

NetworkReconciliationStatus – values are Sent to Network, Network Received Successfully, Network Received with Errors, or Picked Up by Trading Partner.

NetworkReconciliationDateTime – Date and time when the status was set either manually or from the network report.

NetworkErrorMessage – This record is only written if there is an error record in the network report. One record is written for each MS type record in the network report. The value of each reflects the error from the report.

You then use the service in a business process that receives status reports back from the network, so that the additional correlation records are updated with the new network status for each EDI interchange based on Sender and Receiver IDs.

Implementing the Network Report Reconciliation Service

To implement the Network Report Reconciliation Service for use in a business process:

1. Create a Network Report Reconciliation Service Configuration.
2. Configure the Network Report Reconciliation service.
3. Use the Network Report Reconciliation service in a business process.

Configuring the Network Report Reconciliation Service

To configure the Network Report Reconciliation service, you must specify field settings in the GPM:

Field	Description
Config	Name of the service configuration.
action	Action the service should perform. Required. Valid values are: <ul style="list-style-type: none"> ◆ Process data sent ◆ Process network reports

Field	Description
reportType	Type of network report to process. Currently, only one report type is supported: CommerceNetwork. Required if ProcessReports was selected for action field.

Business Process Examples

Example 1: Sending EDI Data

The following example illustrates how the Network Report Reconciliation service might be used in a business process that sends EDI data:

```
<process name="ExampleSendingEDI">
  <sequence name="seq1">
    <!--EDI enveloping would occur before sending so that the proper correlation
      records are written to the database, including InterchangeSenderID,
      InterchangeReceiverID, and InterchangeControlNumber.-->

    <!--send the EDI data to the network using FTP>
    <operation name="FTP">
      <participant name="FtpAdapter"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="xport-ftp-host">sciftp.commerce.stercomm.com</assign>
        <assign to="xport-ftp-port">21</assign>
        <assign to="xport-ftp-mode">PUT</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <!--This service adds additional correlation records to indicate the data
      was sent to the network. Once the reports are received, these additional
      records will be updated to reflect the network status.-->
    <operation name="Network Report Reconciliation">
      <participant name="NetworkReconciliationService"/>
      <output message="ReconcileNetworkReportsInputMessage">
        <assign to="action">ProcessDataSent</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Example 2: Receiving Network Reports

The following example illustrates how the Network Report Reconciliation service might be used in a business process that receives reports from a network:

```
<process name="ExampleReceivingReports">
  <sequence name="seq1">
```



```

<!Receive data format network reports>
<operation name="FTP">
  <participant name="FtpAdapter"/>
  <output message="Xout">
    <assign to="." from="*"></assign>
    <assign to="xport-ftp-host">sciftp.commerce.stercomm.com</assign>
    <assign to="xport-ftp-port">21</assign>
    <assign to="xport-ftp-mode">GET</assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>

  <!Process the reports received and update the correlation records to reflect the
network status.>
  <operation name="Network Report Reconciliation">
    <participant name="NetworkReconciliationService"/>
    <output message="ReconcileNetworkReportsInputMessage">
      <assign to="action">ProcessReports</assign>
      <assign to="reportType">CommerceNetwork</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
</process>

```

Obscure Data - Obscure Primary Document Service

The following table provides an overview of the Obscure Data - Obscure Primary Document service:

System name	ObscureDocument
Graphical Process Modeler (GPM) category	All Services (as ObscureDocument)
Description	The Obscure Data - Obscure Primary Document service masks (obscures) the content of the primary document by replacing the original content with an unintelligible version of the original content. The content of the primary document can be restored to its original form by using the Obscure Data - Reveal Primary Document service.
Business usage	In Application, data processed by a business process (such as the contents of process data and the primary document), while or after it is executed, is normally available in plain text and is easily viewable. You can use the Obscure Data - Obscure Primary Document service to help restrict access to sensitive data, such as passwords, by masking the data into an unintelligible form.
Usage example	Assume that you use Application to process and transmit payroll information to your payroll service provider. You do not want your system management personnel, who are responsible for initiating and monitoring the execution of your business processes, to have access to the data. You can place the sensitive payroll information in the primary document and use the Obscure Data - Obscure Primary Document service to obscure the entire content of the primary document.
Preconfigured?	Yes. A preconfigured instance of the Obscure Data - Obscure Primary Document service named ObscureDocument_Instance is available. There are no configuration parameters and no configuration is required. To use the service, just place ObscureDocument_Instance in a business process.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>This service works in conjunction with the Obscure Data - Reveal Primary Document service.</p> <p>The content of the primary document obscured by the Obscure Data - Obscure Primary Document service can be restored to its original intelligible form with the Obscure Data - Reveal Primary Document service.</p> <p>Note: If you want to obscure the contents of a document as it is being collected from the file system and becomes the primary document, you may be able to use the obscure parameter in the File System adapter. For more information, see <i>File System Adapter</i>.</p>
Application requirements	None
Initiates business processes?	No
Invocation	This service is invoked from within a business process.
Business process context considerations	This service takes the primary document as the input and replaces it with an obscured version of the primary document.

Returned status values	N/A
Restrictions	None
Persistence level	System default
Testing considerations	Test as part of a business process

Example Business Processes

The first BPML example demonstrates the syntax and use of the preconfigured instances of the Obscure Data - Obscure Primary Document service and the Obscure Data - Reveal Primary Document service.

```
<process name="myBusinessProcess">
  <sequence>
    <operation name="ObscureDocument">
      <participant name="ObscureDocument_Instance"/>
      <output message="ObscureDocumentInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
      </input>
    </operation>
```

INSERT OTHER BUSINESS PROCESS OPERATIONS HERE

```
    <operation name="RevealDocument">
      <participant name="RevealDocument_Instance"/>
      <output message="RevealDocumentInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
```

INSERT OTHER BUSINESS PROCESS OPERATIONS HERE

```
  </sequence>
</process>
```

The next example uses both the preconfigured instances of the Obscure Data - Obscure Primary Document service and the Obscure Data - Reveal Primary Document service and the File System adapter to obscure and unobscure data. For more information about the File System adapter, see *File System Adapter*.

```
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
```

```
  <!-- Unobscure the collected document, using the Obscure Data - Reveal
        Primary Document service, to extract information from it -->
```

```
    <operation name="RevealDocument">
      <participant name="RevealDocument_Instance"/>
      <output message="RevealDocumentInputMessage">
```

```

    <assign to="." from="*"></assign>
</output>
<input message="inmsg">
    <assign to="." from="*"></assign>
</input>
</operation>

<!-- Extract the desired information from the unobscured document and use
it as needed in the business process -->

<!-- Reobscure the document using the Obscure Data - Obscure Primary
Document service -->

<operation name="ObscureDocument">
    <participant name="ObscureDocument_Instance"/>
    <output message="ObscureDocumentInputMessage">
        <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
        <assign to="." from="*"></assign>
    </input>
</operation>

<!-- Perform other business process operations, as needed -->

<!-- Extract an unobscured version of the document to the file system using
the File System adapter -->

<operation name="File System Adapter">
    <participant name="FSA_Outbound"/>
    <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="assignedFilename">%^.txt</assign>
        <assign to="assignFilename">>true</assign>
        <assign to="collectionFolder">/dir_path
                                dir_path
    </output>
</operation>

```

Obscure Data - Process Data Values Service

The following table provides an overview of the Obscure Data - Process Data Values service:

System name	ObscureParameter
Graphical Process Modeler (GPM) category	All Services (as ObscureParameter)
Description	Each instance of the Obscure Data - Process Data Values service can be used to insert up to five pre-configured parameter name-value pairs into process data. The values associated with each of the parameter names are masked by replacing the original content with an unintelligible version. The unobscured plain text value can be retrieved by using the revealObscured XPath function.
Business usage	In Application, the data processed by a business process (contents of process data, primary document), as it is being executed or after it has finished executing is readily available in plain text and easily viewable. The purpose of the Obscure Data - Process Data Values service is to help restrict access to sensitive data, such as passwords, etc., by masking them and converting the values to unintelligible ones. If you need to obscure the entire content of the primary document, use the Obscure Data - Obscure Primary Document service and if you only need to obscure certain elements in process data, use the Obscure Data - Process Data Values service.
Usage example	Assume that a business process has been set up to route information based on the information stored in a secure repository. The security credentials (user ID, password, etc.) required to access the secure repository could be obscured while configuring the Obscure Data - Process Data Values service and retrieved only within the adapter used to establish a connection to the secure repository.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	This service is designed to work in conjunction with the RevealObscured XPath function. See <i>revealObscured XPath Function</i> on page 1007 for more information.
Application requirements	The parameter name-value pairs to be inserted into process data must be specified while configuring the service instance in Application, in order to prevent unauthorized viewing of sensitive information in the Graphical Process Modeler (GPM) or the BPML code.
Initiates business processes?	No
Invocation	This service is invoked from within a business process.
Business process context considerations	This service inserts the configured, and obscured, parameters into process data. It does not affect the primary document.
Returned status values	<ul style="list-style-type: none">◆ Success◆ Error
Restrictions	None

Persistence level	System default
Testing considerations	None

Implementing the Obscure Data - Process Data Values service

To implement the Obscure Data - Process Data Values service, complete the following tasks:

1. Create an Obscure Data - Process Data Values service configuration. For information, see *Managing Services and Adapters*.
2. Configure the adapter. For information, see *Configuring the Obscure Data - Process Data Values service* on page 1006.
3. Create and enable a business process that includes the Obscure Data - Process Data Values service.
4. Test the business process and the service.
5. Run the business process.

Configuring the Obscure Data - Process Data Values service

To configure the Obscure Data - Process Data Values service, you must specify field settings in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Parameter Name	<p>Name of the obscured parameter. Obscure Data – Process Data Values service uses the value you specify for this parameter as the element name in the ProcessData of the business process.</p> <p>This service creates process data elements based on the name-value pairs.</p>

Field	Description
Parameter Value	<p>Value of the obscured parameter. Obscure Data - Process Data Values service encrypts the value you specify for this parameter. The service uses the encrypted representation for the value in ProcessData with the Parameter Name for the element. The following example illustrates the encrypted value in an example of process data:</p> <pre><ProcessData> <admin>AAAAEQAAAAUAAAAeAAAAAGgAAABkAAAAOAAAAFgAAAAk=</admin> <support>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA==</support> <ftpdemo>AAAAFgAAABUAAAAADAAAAFwAAABIAAAACAAAAHQAAAAI=</ftpdemo> </ProcessData></pre> <p>This service creates process data elements based on the name-value pairs.</p> <p>Note: The value you specify for this parameter must comply with the XML naming rules for XML elements and XML syntax rules. For more information on XML conventions, see www.w3.org/XML/.</p>

Note: A maximum of five of the above name-value pairs can be specified per instance of the Obscure Data - Process Data Values service.

This service creates process data elements based on the name-value pairs provided during configuration. No Obscure Data - Process Data Values service parameters are configured with the GPM. To retrieve the unobscured parameter values, use the revealObscured XPath function.

revealObscured XPath Function

The revealObscured XPath function is used to retrieve values associated with the obscured parameter names, in plain text form.

The syntax and an example demonstrating the usage of the revealObscured XPath function is shown below:

Syntax:

Name of Obscured Parameter

Example:

obscuredPasswordParameterName

The revealObscured XPath function can be used in an assign statement in a service invocation. For example, if a service requires a password passed from the workflow context, you could enter this password using the Obscure Data - Process Data Values service to prevent a plain text password from being passed in the business process. The following assign statement could be then used to pass the plain text password to the service.

```
<assign to="password" from="revealObscured(ParameterName)" />
```

Example Business Processes

The first example uses a single parameter in the Obscure Data - Process Data Values service to take a user's password, obscure it and add it to process data. In this case, the User ID is specified in the Parameter Name field and the associated password is specified in the Parameter Value field. See *Configuring the Obscure Data - Process Data Values service* on page 1006 for more information. Then, when needed in the business

process, the revealObscured XPath function is used to retrieve the obscured password from process data in a plain text (unobscured) form. The User ID and password can then be used to perform some function, for example, access a system folder.

```
<process name="Obscure_Data_Process_Data_Values_Example">
  <sequence>

    <!-- Use the Obscure Data - Process Data Values service to add the obscured
         password to process data for later retrieval and use in the business
         process -->

    <operation name="ObscureParameter">
      <participant name="Example_Obscure_Param" />
      <output message="ObscureParameterInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    INSERT OTHER BUSINESS PROCESS OPERATIONS HERE

    <!-- Retrieve the obscured password from process data and unobscure it
         using the revealObscured XPath function -->

    <assign to="RevealedPassword" from="revealObscured(User ID
```

The next example uses the Obscure Data - Process Data Values service to obscure both the User ID and password and add them to process data. In this example, there will be two parameters specified. For the first parameter, **UserID** is specified in the Parameter Name field and the actual User ID is specified in the Parameter Value field. For the second parameter, **Password** is specified in the Parameter Name field and the password associated with the User ID is specified in the Parameter Value field. Then, when needed, the revealObscured XPath function is used to retrieve the two parameters from process data in a plain text (unobscured) form. The unobscured User ID and password are then available to be used as needed in the business process.

Obscure Data - Reveal Primary Document Service

The following table provides an overview of the Obscure Data - Reveal Primary Document service:

System name	Reveal Document
Graphical Process Modeler (GPM) category	All Services (as RevealDocument)
Description	The Obscure Data - Reveal Primary Document service is used to restore primary document content previously obscured using the Obscure Data - Reveal Primary Document service to a plain text form.
Business usage	<p>In Application, data processed by a business process (such as the contents of process data and the primary document), while or after, it is executed is normally available in plain text and is easily viewable. You can use the Obscure Data - Obscure Primary Document service to help restrict access to sensitive data, such as passwords, by masking the data into an unintelligible form.</p> <p>You will use the Obscure Data - Reveal Primary Document service to restore the previously obscured content of the primary document to a plain text form.</p>
Usage example	Assume that you use Application to process and transmit payroll information to your payroll service provider. To prevent your system management personnel from having access to it, your business process placed the sensitive payroll information in the primary document and used the Obscure Data - Obscure Primary Document service to obscure the entire content of the primary document. You will use the Obscure Data - Reveal Primary Document service to restore the information to a plain text version in the primary document, before transmitting it to the payroll service provider.
Preconfigured?	Yes. A preconfigured instance of the Obscure Data - Reveal Primary Document service named <code>RevealDocument_Instance</code> is available. There are no configuration parameters and no configuration is required. To use the service, just place <code>RevealDocument_Instance</code> in a business process.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	<p>This service is designed to work in conjunction with the Obscure Data - Obscure Primary Document service.</p> <p>The content of the primary document obscured by the Obscure Data - Obscure Primary Document service can be restored to its original intelligible form with the Obscure Data - Reveal Primary Document service.</p> <p>Note: If you want to unobscure the contents of the primary document while extracting it to the file system, you may be able to use the <code>unobscure</code> parameter in the File System adapter. For more information, see <i>File System Adapter</i>.</p>
Application requirements	None
Initiates business processes?	No
Invocation	This service is invoked from within a business process.

Business process context considerations	This service takes the obscured primary document as the input and replaces it with a plain text version of the primary document.
Returned status values	N/A
Restrictions	None
Persistence level	System default
Testing considerations	Test as part of a business process

Example Business Process

The following BPML demonstrates the use of the preconfigured instances of the Obscure Data - Obscure Primary Document service and the Obscure Data - Reveal Primary Document service.

For more examples of obscuring and unobscuring the primary document, see the examples in *Obscure Data - Obscure Primary Document Service*.

OdetteFTP Adapter

The following table provides an overview of the OdetteFTP adapter:

Note: Use the OdetteFTP adapter instead of the OFTP adapter.

System name	OdetteFTP Adapter
Graphical Process Modeler (GPM) category	All Services
Description	<p>This adapter is used to exchange files with trading partners using the OdetteFTP (OFTP) protocol. There are two modes of operation:</p> <ul style="list-style-type: none">◆ Mailbox Mode - Files are sent from and placed into a mailbox, which provides more flexibility, better performance, and easier maintenance. Requires a separate license and special settings in the partner profile.◆ Manual Mode - Basic functionality. Files must be sent or received through a business process.
Business usage	<p>The OdetteFTP adapter supports point-to-point communication between trading partners using the OFTP protocol version 1.2, version 1.3, or version 1.4 over ISDN or TCP/IP as an underlying communication protocol. The OdetteFTP adapter implements the complete OFTP and CAPI/IP protocol stack.</p>

Usage example	<p>For both Mailbox and manual modes of operation, the sequence of events is:</p> <ol style="list-style-type: none"> 1 Trading Partner A requests a session with Trading Partner B. 2 Trading Partner A sends the Start Session Ready Message command, indicating that the application is communicating using OFTP. 3 Trading Partner A requests permission to send a file. The request contains information such as the origin and destination of the file, its name, and physical size. 4 Trading Partner A sends the actual file to Trading Partner B. 5 Immediately after the file is transferred, Trading Partner A sends the End of File Identification command, which contains control totals to ensure the integrity of the sent file. 6 When the file reaches its destination, Trading Partner B returns an End-to-End Response command to Trading Partner A to inform them that the file has been received. 7 Trading Partner B issues an End of Session Identification (ESID) command to request that the communication session be terminated. 8 Trading Partner A terminates the session. <p>Mailbox Mode</p> <ul style="list-style-type: none"> ◆ Initiator Role - The OdetteFTP adapter is called from a business process with parameter PhysicalPartnerContract. The adapter looks up the mailbox of the remote partner and sends all messages in the mailbox to the Remote Physical Partner. Messages can be received in the same session. ◆ Responder Role - Incoming messages are stored in the partner mailbox. The file can be moved to an archive mailbox. A remote partner can initiate a session and poll for messages. <p>Manual Mode</p> <ul style="list-style-type: none"> ◆ Initiator Role - A set of documents can be sent to the remote partner. Files from the remote partner can be received in the same session. A business process is initiated for received files and End to End ResPonses (EERPs) and Negative End ResPonses (NERPs). ◆ Responder Role - Files from a remote partner can be received and messages can be sent to the remote partner in the same session.
Preconfigured?	<p>No.</p> <p>For manual mode, there is an OdetteFTP adapter instance named OFTPSendFile which is used for the preconfigured business processes oftpin and oftpout. Configure the OFTPSendFile instance prior to using it.</p>
Requires third party files?	<p>No, the OdetteFTP adapter implements the complete OFTP and CAPI/IP protocol stack.</p>
Platform availability	<p>All platforms supported by Application</p>
Related services	<p>OdetteFTP Scheduler service (Mailbox mode only) OdetteFTP Queue Handler service (Mailbox mode only)</p>

Application requirements	<p>For all modes and communication protocols:</p> <ol style="list-style-type: none"> 1 Configure information about Application and the partner OFTP server in the Partner Profile XML File. Some parameters depend on a bilateral agreement between you and your Partner (default directory is <install_dir>/install/properties/PartnerProfile.xml). 2 Create an OdetteFTP adapter instance using the preconfigured instance named OFTPSendFile as a template. <p>For ISDN Communication only:</p> <ul style="list-style-type: none"> ◆ Install the hardware ISDN Router (Brick) <p>For operation in Mailbox Mode:</p> <ol style="list-style-type: none"> 1 Create a Application User Account (Accounts > User Accounts). 2 Create a partner mailbox with a sub-mailbox named Inbox (Deployment > Mailboxes > Configuration). 3 Create a Virtual Root (Deployment > Mailboxes > Virtual Roots) for this mailbox. 4 Create a Time Schedule within the corresponding PhysicalPartnerContract in the Partner Profile to schedule the initiating of OFTP sessions. Enter the name of the initiator business process into the XML field <TimeScheduleTable>/<InitiatorBusinessProcess>. 5 Enable the OFTPScheduler service. 6 Create an initiator business process that passes the PhysicalPartnerContract parameter to the OdetteFTP instance. 7 Use the Odette FTP Queue Handler to queue one or more files for sending. <p>For operation in manual mode:</p> <ul style="list-style-type: none"> ◆ Create a business process for sending files using the sample business process <i>oftpout</i> as a template for manual Mode.
Initiates business processes?	<p>Mailbox Mode - The message is stored in the Partner Mailbox directly or is taken from the Partner Mailbox.</p> <p>Manual Mode - The OdetteFTP adapter initiates business processes when an OFTP virtual file is received.</p>
Invocation	<p>Mailbox Mode - The OdetteFTP adapter is initiated by the OdetteFTP Scheduler service based on a partner-specific schedule. It can also be initiated by inbound communication from a trading partner.</p> <p>Manual Mode - The OdetteFTP adapter can be initiated from a business process by passing an OFTPDataSet XML structure.</p>
Business process context considerations	<p>Mailbox Mode - The adapter takes the Physical Partner Contract name as a BPML input parameter. All messages queued by the OFTP Queue Handler in the Partner Mailbox for sending, are sent.</p> <p>Manual Mode - The adapter takes one or more Primary Documents within the OFTPDataSet XML Input structure from the workflow.</p>
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error

Restrictions	<p>The OdetteFTP adapter has the following restrictions:</p> <ul style="list-style-type: none"> ◆ OFTP version 2.0 is not supported (only OFTP 1.2, version 1.3, and version 1.4 is supported). ◆ Specific ISDN hardware is required for ISDN communication only. Only ISDN routers from Funkwerk Enterprise Communications GmbH (formerly Bintec) with Remote CAPI functionality are supported. ◆ Usage of Application Mailbox system requires a Mailbox license. ◆ Limited support for VAN scenario. Pass-through mode is not supported. ◆ Special Logic is not supported. ◆ Restarting is not supported.
Persistence level	The adapter has no pre-set persistence level.
Testing considerations	<p>For testing ISDN, you need a system handling the OFTP Partner role. For CAPI/ISDN communication, a BINTEC ISDN Router (Brick) is required.</p> <p>In a loop back scenario, Application can be configured to take over the role of both an OFTP Initiator and an OFTP Responder.</p>

Prerequisite

You must be familiar with the Odette File Transfer Protocol and ISDN or TCP/IP protocol.

How the OdetteFTP Adapter Works

The OdetteFTP adapter supports point-to-point communication between trading partners using the OFTP version 1.2, version 1.3 or version 1.4 protocol over ISDN or TCP/IP as an underlying communication protocol for the following file types:

- Unformatted (binary)
- Text (for example, ASCII)
- Fixed length
- Variable-length

The OdetteFTP adapter implements the complete OFTP and CAPI/IP protocol stack. The adapter can handle incoming and outgoing ISDN calls, EERP (End to End Response), and can auto-generate EERPs for incoming files.

In Mailbox mode, put one or more FILE entries into a process queue and the partner mailbox using the OdetteFTP Queue Handler service. When the OdetteFTP adapter is initiated by the OFTP Scheduler, it searches the local partner Inbox mailbox for files to send and initiates an OFTP session. Files, EERPs, and NERPs (Negative End Responses) can be received in the same session. When the OdetteFTP Adapter receives files, they are stored in the root of the partner mailbox. For incoming EERPs and NERPs the file entry is updated and a new entry is created for the confirmation.

In manual mode, for each incoming transmission, a business process is initiated to move the transmission through Application. For each EERP and NERP, a business process is started. The process data of the initiated business process contains all parameters required to process the inbound call (file properties or

EERP/NERP properties). For outgoing transmissions, the OdetteFTP adapter writes the send status of all files in one transmission (OFTPDataSet) into process data.

Note: Application supports logging, tracing, and debugging. The adapter supports log level `COMMTRACE` which can be used if `INFORMATION` does not provide enough details about the OFTP commands sent. Be careful when using log level `DEBUG` or `ALL`, because the amount of information can fill disk space quickly. These functions are for testing only.

Sample Business Scenarios

1. You have many files to send each day. You can send them in one transmission (OFTP session) to your partner by putting all files into the partner's Inbox (Mailbox mode) or by passing more than one `DataItem` (document) in the OFTPDataSet XML structure (manual mode). This is much more efficient than initiating an OFTP session for each single file.

2. You want to place just-in-time orders to a trading partner. In Mailbox mode, set up a simple or complex schedule for when you want to send files to your partner.

In manual mode, you can set up a business process by using the preconfigured business process `oftpout` as a template. Add your information to the preconfigured business process `oftpin` for handling inbound EERPs, NERPs, and files from your partner.

3. You want to receive files from your partner. In Mailbox mode, set up a simple or complex schedule when you want to poll a partner's OFTP mailbox for files. Or, you can receive files when the partner initiates the OFTP session. A business process for incoming files is not required in Mailbox Mode.

In manual mode, configure the business process `oftpin` to initiate in the adapter configuration. Create a business process with the same name or use the preconfigured business process `oftpin` as a template for handling inbound files.

4. To send a file to a trading partner, create an instance of the OdetteFTP adapter, configure all parameters and activate the adapter.

In Mailbox Mode, queue the file and pass the `PhysicalPartnerContract` name to the adapter. The status of each file is created and updated in the process history.

In manual mode, send an OFTPDataSet XML structure to it which contains one or more `DataItems` (documents) for sending. The business process remains in `WAIT_ON_IO` status until all files have been sent. Then, the adapter reactivates the process and writes an OFTPResponse XML structure into the process data. The response contains status information for each file which can be handled in the business process afterwards. The status of the adapter call is `Success` if all files have been sent successfully; otherwise the status is `Error`.

Note: The `WAIT_ON_IO` state avoids blocking thread resources when transferring data.

Implementing the OdetteFTP Adapter in Mailbox Mode

To implement the OdetteFTP adapter in Mailbox mode:

1. Edit the default **PartnerProfile.xml** file in `install_dir/install/properties` and add your OFTP partner information. For each partner you need to add a separate `PhysicalPartnerContract` and

LogicalPartnerContract including the partner definitions referenced in the contracts. Use the template definitions in the **PartnerProfile.xml** file as a starting point.

Note: There is only one PartnerProfile.xml file for all OdetteFTP adapter instances, and it is also used for the OdetteFTP Queue Handler and the OdetteFTP Scheduler.

2. After saving the file, encrypt the passwords in the PartnerProfile.xml file.
3. Create user accounts, virtual roots, and Mailboxes with Inboxes for you and all your partners.
4. Activate your license for the OdetteFTP adapter and for the Mailbox.
5. Create an OdetteFTP adapter configuration.
6. Configure the adapter.
7. After saving the adapter instance configuration, the PartnerProfile.xml file is loaded by the new adapter instance. The file must be completed before clicking Save in the adapter instance configuration. As soon as the instance is saved, the PartnerProfile.xml file is loaded.

To activate changes in the Partner Profile when other adapter instances are already running, first stop all adapter instances and the Odette FTP Scheduler. Restart them after applying the change.

Note: Check the OdetteFTP log file before using the adapter. Verify that the PartnerProfile.xml file has been loaded without errors. Otherwise, the adapter is not ready to use.

8. Create a business process that passes the PhysicalPartnerContract parameter to the adapter.
9. Test the business process and the adapter.
10. Enable the OFTP Scheduler service if you have created a Time Schedule in the Partner Profile.
11. Use the OdetteFTP Queue Handler service to send messages.

Implementing the OdetteFTP Adapter in Manual Mode

To implement the OdetteFTP adapter in manual mode:

1. Edit the default **PartnerProfile.xml** file in *install_dir/install/properties* and add your OFTP partner information. For each partner you need to add a separate PhysicalPartnerContract and LogicalPartnerContract including the partner definitions referenced in the contracts. Use the template definitions in the **PartnerProfile.xml** file as a starting point.

Note: There is only one PartnerProfile.xml file for all OdetteFTP adapter instances.

2. After saving the file, encrypt the passwords in the PartnerProfile.xml file.
3. Activate your license for the OdetteFTP adapter.
4. Create an OdetteFTP adapter configuration.
5. Configure the adapter.
6. After saving the adapter instance configuration, the PartnerProfile.xml file is loaded by the new adapter instance. The file must be completed before clicking Save in the adapter instance configuration. As soon as the instance is saved, the PartnerProfile.xml file is loaded.

To activate changes in the Partner Profile when other adapter instances are already running, first stop all adapter instances. Restart them after applying the change.

Note: Check the OdetteFTP log file before using the adapter. Verify that the PartnerProfile.xml file has been loaded without errors. Otherwise, the adapter is not be ready to use.

7. Create and enable a business process that includes the OdetteFTP adapter. You can use the sample business processes (oftpin and oftput) and modify them to suit your installation.
8. Create the OFTP Data Set XML structure in process data with the appropriate parameters so it can be passed to the adapter.
9. Test the business process and the adapter.

Configuring the OdetteFTP Adapter

To configure the OdetteFTP adapter, specify field settings in Application or the GPM:

Field	Description
Name	Unique and meaningful name for the adapter configuration, for reference purposes. Required.
Description	Meaningful description for the adapter configuration. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Communication Mode	The mode of communication for this instance. Required. Valid values are: <ul style="list-style-type: none">◆ IP◆ ISDN
Inbound Business Process Name	The name of the business process initiated for received files. Required for manual mode only.
User	For manual mode, the Application user ID to use with the business process. Required for manual mode. Type the user ID, or select a user ID from the list. Valid value is any valid Application user ID. Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.

For Communication Mode of IP

Field	Description
Perimeter Server	Name of the Perimeter Server or Perimeter Server group. Required.

Field	Description
IP Server Listener Port	TCP/IP port number for incoming IP calls. Optional. Default is 3305.
Accept inbound calls without ACL check	Whether to verify inbound calls with the access control list. Default is Yes. Required.

For Communication Mode of ISDN

Field	Description
ISDN Router Address	The name or IP address of the ISDN router (brick). Required.
ISDN Router Port	The port for the ISDN router. The adapter uses this port to open up an IP-socket to the router. Required.
Controller Number	Select the controller number of your ISDN router which should be used for incoming calls. Required.
Keep Alive Interval (in seconds, 0 = unlimited)	Maximum length of time (in seconds) to wait for the response from a finished transmission. This is the time that a business process will stay in consume mode. Default is 2. Required. Valid values include any positive integer between 0 and 9999999999. Specify 0 for unlimited wait time.
Number of Channels	The number of channels supported by the ISDN router. See your ISDN router manual to obtain the number of supported channels. Default is 2. Required.
Accept Local MSN	Whether to accept incoming calls from a partner for the ISDN MSN specified only (fill in the phone number) or accept all MSN (parameter left blank). Optional.
CAPI Socket Timeout	The number of seconds before the CAPI socket times out. Default is 2. Required.
Accept inbound CAPI calls without ACL check	Specifies whether you want to accept inbound CAPI calls with or without check the settings in the Access Control List of the Partner Profile. Valid values are Yes (accept calls without ACL check) and No. Default is Yes. Required.
Accept inbound CAPI calls without Calling Partner Address	Specifies whether you want to accept inbound CAPI calls with or without the "Calling Party Address" field populated. Valid values are Yes (accept calls without the Calling Party Address field populated) and No. Default is Yes. Required.

Example Business Process for Mailbox Mode

This example shows a business process that sends one message to a partner:

Partner Mailboxes

To use OFTP with the mailbox system, you must have a standard sub-mailbox layout for storing scheduled and received OFTP files in your Mailbox Virtual Root. Files to be sent are stored in the sub-mailbox named Inbox. Received files are stored in the Virtual Root of the Mailbox User.

Parameter Passed from Business Process to Adapter - Mailbox Mode Only

Field	Description
PhysicalPartnerContract	Name of the Physical Partner Contract as defined in the Partner Profile. The contract is used to look up the remote partner mailbox.

Error Messages

The following error messages will be only displayed in the Status Report for the instance of the OdetteFTP adapter.

Advanced Status	Description
OFTP_INITIALIZATION_FAILED	An error occurred during initializing of required components. For example, the Service Framework providing logging services did not start successfully.
OFTP_SEND_FAILED	The adapter was not able to send the messages passed in the OFTPDataSet structure successfully. Details are given in the Status Report.
DATASET_NOT_COMPLETE	In Manual Mode, the OFTPDataSet structure passed to the adapter was parsed and did not contain all required fields (LogicalPartnerContract, PhysicalPartner Contract).
OFTP_FAILURE	A general error occurred during processing that does not fit into the categories listed above. The reason is noted in the status report. Note: If the adapter couldn't be started successfully, such as a wrong configuration, then check the OdetteFTP log for error details. The Partner Profile configuration may have errors.
WRONG_WF_PARAM	In Mailbox Mode only, either the PhysicalPartnerContract is not found or the MailboxUser is not defined in the contract.

Business Process Configuration - Manual Mode Only

You are provided with sample business processes for manual mode (**oftpin** for inbound transmissions and EERP/NERP notifications and **oftpout** for initiating a session to a partner) that you can modify to use with the OdetteFTP adapter. Details on how to modify these business processes are given in XML comments in the preconfigured business processes. The preconfigured business processes require the OdetteFTP adapter instance OFTPSendFile to be configured and enabled. There are no parameters to be configured in the GPM.

OFTP Data Set XML Structure - Manual Mode Only

The OdetteFTP adapter is able to send one or multiple Application messages to one physical partner within a single adapter call. The OFTP Data Set XML structure has to be created in the process data and passed to the adapter, as in the following example:

```
< FileFormat>[ U|T|V|F]</ FileFormat>
<OFTPFileUserField>free_user_content</ OFTPFileUserField>
<RecordDelimiter>one_or_two_delimiters<RecordDelimiter>
</properties>
<document index="1">
  <PrimaryDocument SCIOBJECTID="document_id_1"/>
</document>
</DataItem_1>
Optional:
<DataItem_n>
...
</DataItem_n>
</OFTPDataSet>
```

Defining XML Node Name Parameters - Manual Mode Only

To define the XML node name parameters:

1. Edit the PartnerProfile.xml file at *install_dir/install/properties* if you are using the default partner profile.
2. Make the following changes to the file.

The following table describes the parameters that need to match the definitions in the OFTP partner profile. Make sure that all required parameters belonging to your logical and physical contract are

configured correctly, including all parameters of the logical and physical partners referenced in the contract part. Details are described in the default partner profile in XML comments.

Parameter	Description
OFTPDataSet@Physical PartnerContract	The unique name of the physical partner contract as defined in the Partner Profile XML file. This is an 80-character string. Required.
LogicalPartnerContract	The unique name of the logical partner contract as defined in the Partner Profile XML file. This is an 80-character string. Required.
OFTPVirtualFilename	The OFTP Virtual File name. Defined according to the bilateral agreement with your trading partner. This is a 26-character string. Optional. If omitted a default file name is taken from the partner profile contract. Note: The virtual file name, date, and time are used to uniquely define a file.
Date	The date tag used to send the message. This is a six-digit or eight-digit number. Format is as follows: <ul style="list-style-type: none"> ◆ YYMMDD (used for version 1.2 and version 1.3) ◆ YYYYMMDD (used for version 1.4) Optional. Note: The virtual file name, date, and time are used to uniquely define a file.
Time	The time stamp from when a file is made available for transmission at the sender's location. This is a six-digit or ten-digit number. Format is as follows: <ul style="list-style-type: none"> ◆ HHMMSS (used for version 1.2 and version 1.3) ◆ HHMMSS<four-digit counter> (used for version 1.4) Optional, but it is recommended that you specify the date and time stamp when the application created the file. Note: The virtual file name, date, and time are used to uniquely define a file.
FileFormat	This field specifies the format of the virtual file. Valid values are: U - unstructured binary file T - text file F - fixed-length record binary file V - variable-length record binary file Optional.
OFTPFileUserField	Used as defined by your bilateral agreement with your partner. Optional.
RecordDelimiter	One or two characters which are used to separate records if the file format is V (variable length). For example, 10 indicates a line feed or 13, 10 indicates a carriage return and line feed. This is a four-character string. Optional.

Example of the OFTPResponse XML Structure - Manual Mode Only

For each OFTP Data Set request, a response structure is created in the process data which contains the process results of the OdetteFTP adapter call. The following example shows a sample response structure:

```
<OFTPResponse PhysicalPartnerContract="physical_contract_name">
  <DataItem_1>
    <Status> [Success | Failure | Skipped] </Status>
  </DataItem_1>
</OFTPResponse>
```

```

    <Reason>The long description</Reason>
    <ReasonCode>two_digit_reason_code</ReasonCode>
    <Retry>[Yes|No]</Retry>
  </DataItem_1>

  Optional:
  <DataItem_n>
    ...
  </DataItem_n>
</OFTPResponse>

```

The following table describes the parameters of the response structure:

Parameter	Description
OFTPResponse@PhysicalPartnerContract	The unique name of the physical partner contract as defined in the Partner Profile XML file. This is an 80-character string.
Status	The status of the response structure. This is a string. Valid values are: <ul style="list-style-type: none"> ◆ Success ◆ Failure ◆ Skipped
Reason	This is a long error description. This is a string.
ReasonCode	The error reason code as defined by the OFTP specification.
Retry	If an error occurs, the field specifies whether the virtual file should be resent. Valid values are Yes and No.

OFTP Inbound XML Structure - Manual Mode Only

For each single file received from a partner, a business process is initiated with the following file description in the process data:

```

<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <PrimaryDocument SCIOBJECTID="unique_document_id"/>
  <OFTPInbound>
    <Type>FILE</Type>
    <FileName>virtual_file_name</FileName>
    <FileSize>1</FileSize>
    <Originator>originator_name</Originator>
    <Destination>destination_name</Destination>
    <Time>time</Time>
    <Date>date</Date>
    <FileFormat>[U|T|V|F]</FileFormat>
  </OFTPInbound>
</ProcessData>

```

For each OFTP EERP and NERP notification, a business process is initiated containing following OFTPInbound structure in process data:

EERP

```
<OFTPInbound>
  <Type>EERP</Type>
  <FileName>virtual_file_name</FileName>
  <Originator>originator_name</Originator>
  <Destination>destination_name</Destination>
  <Time>time</Time>
  <Date>date</Date>
</OFTPInbound>
```

NERP

```
<OFTPInbound>
  <Type>NERP</Type>
  <FileName>virtual_file_name</FileName>
  <Originator>originator_name</Originator>
  <Destination>destination_name</Destination>
  <Creator>NERP_creator</Creator>
  <Time>time</Time>
  <Date>date</Date>
  <Reason><Reason>
</OFTPInbound>
```

An NERP notification contains two additional fields:

Reason - Why the partner rejected the file on application level.

Creator - Specifies the creator of the NERP, which may be different from the Destination.

The following table describes the parameters of the OFTP inbound XML structure:

Parameter	Description
FileName	This is the OFTP virtual file name as defined by the bilateral agreement with your partner. Note: The virtual file name, date, and time are used to uniquely define a file.
FileSize	The amount of space used at the originator to store the virtual file. The size includes user data only and is specified in K (1024) bytes. The maximum is 9999999.
Date	The date tag used to send the message. This is a six-digit or eight-digit number. Format is as follows: <ul style="list-style-type: none">◆ YYMMDD (used for version 1.2 and version 1.3)◆ YYYYMMDD (used for version 1.4) Note: The virtual file name, date, and time are used to uniquely define a file.
Time	The time stamp from when a file is made available for transmission at the sender's location. This is a six-digit or ten-digit number. Format is as follows: <ul style="list-style-type: none">◆ HHMMSS (used for version 1.2 and version 1.3)◆ HHMMSS<four-digit counter> (used for version 1.4) Note: The virtual file name, date, and time are used to uniquely define a file.
Originator	Identifies the sender of the virtual file. This is the location that mapped the data for transmission.

Parameter	Description
Destination	Identifies the final recipient of the transmission. This is the location that looks into the virtual file content.
FileFormat	This field specifies the format of the virtual file. Valid values are: U - unstructured binary file T - text file F - fixed-length record binary file V - variable-length record binary file

Example Business Process for Manual Mode

This example shows a business process that sends one message to a partner. This example assumes that the Application document is located in the process data root under **/doc1**.

Note: This business process is a template for manual mode operation.

```
<process name="oftpsend">
  <sequence name="oftp">

    <operation name="CreateOFTPDataSetStructure">
      <participant name="AssignService"/>
      <output message="fromProcessData">
        <assign to="OFTPDataSet/@PhysicalPartnerContract"
          from="'Sterling_VW1'"></assign>
        <assign to="OFTPDataSet/DataItem_1/document" from="doc1/node()"></assign>
        <assign to="OFTPDataSet/DataItem_1/document/@index" from="'1'"></assign>
        <assign to="OFTPDataSet/DataItem_1/properties/LogicalPartnerContract"
          from="'SterlingAndVW'"></assign>
        <!-- Add optional parameters here, if used -->
        <assign to="." from="*"></assign>
      </output>
      <input message="toProcessData">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <!-- Start OFTP send process -->

    <operation name="SendOFTP">
      <participant name="OFTPSendFile"/>
      <output message="Out">
        <assign to="." from="*"></assign> <!-- Pass OFTPDataSet -->
      </output>
      <input message="In">
        <assign to="." from="*"></assign> <!-- Get OFTPResponse -->
      </input>
    </operation>

  </sequence>
</process>
```

OdetteFTP Queue Handler Service

The following table provides an overview of the OdetteFTP Queue Handler service:

System name	OdetteFTP Queue Handler Service
Graphical Process Modeler (GPM) category	All Services
Description	The OdetteFTP Queue Handler is used to queue a file for sending by adding entries to the Mailbox and a process queue. These entries contain all information required by the OdetteFTP adapter to identify the Physical Partner Contract and prepare a file for sending, the Application document or MessageID, Logical Partner Contract, and OFTP file name.
Business usage	Use the OdetteFTP Queue Handler to put a file into the send queue for a partner. The send queue includes all the File, End to end Response (EERP) and Negative End Response (NERP) items in the process history with a status that indicates the item needs to be sent or resent. The file can be a document in a business process or a message in the partner mailbox. If it is a document, it will be stored in the partner mailbox automatically.
Usage example	<p>There is a preconfigured instance of the OdetteFTP Queue Handler service named OFTPQueueHandler. Three business process templates are provided to put items in the send queue:</p> <ul style="list-style-type: none">◆ oftpfile: Put a file into the OFTP queue◆ oftpeerp: Put an EERP into the OFTP queue◆ oftpnerp: Put a NERP into the OFTP queue
Preconfigured?	Yes - OFTPQueueHandler
Requires third party files?	No
Platform availability	All supported platforms for Application
Related services	OdetteFTP Scheduler service OdetteFTP adapter (Mailbox mode only)
Application requirements	The OdetteFTP Queue Handler service is used in conjunction with the OdetteFTP adapter in Mailbox mode.
Initiates business processes?	No
Invocation	This service is invoked from a business process.

Business process context considerations	<p>All messages queued by the OdetteFTP Queue Handler service in the Partner Mailbox will be sent. A message can be queued for sending in two different ways:</p> <ul style="list-style-type: none"> ◆ The user already has a message in the partner mailbox. In this case, start the OFTPQueueHandler and provide the MessageID in the DataItem-File-XML Structure without specifying a primary document. ◆ The user has a primary document in the business process, which is not stored in a partner mailbox. In this case, the OFTPQueueHandler is started with the Primary Document link in the DataItem-File-Structure without specifying a MessageID. The document will be stored in the partner mailbox automatically. <p>In both cases, you must specify the Logical Partner Contract, which is used by the OdetteFTP adapter to identify the related OFTP send parameters defined in the Physical Partner Contract.</p>
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	None
Persistence level	System default
Testing considerations	Not applicable

Configuring the OdetteFTP Queue Handler Service

There are no configurable parameters for the OdetteFTP Queue Handler service. To use the OdetteFTP Queue Handler service:

1. Create an OdetteFTP Partner Profile.
2. Create the following XML structures in a business process.
3. Pass the XML structures to the OdetteFTP Queue Handler service from the business process.

Input XML Structures

To put files, EERPs or NERPs into the OFTP queue, the OdetteFTP Queue Handler service requires the following XML input structures which have to be passed to the Service from a business process:

XML Input Structure for Message Type FILE:

```

<DataItem>
  <FILE>
    <properties>
      <LogicalPartnerContract>log_partner_name</LogicalPartnerContract>
      <!-- LogicalPartner Properties overriding defaults in partner profile -->
      <OFTPVirtualFilename>virtual_filename</OFTPVirtualFilename>
      <Date>date</Date>
      <Time>time</Time>
      <FileFormat>[ U|T|V|F]</FileFormat>
      <OFTPFileUserField>free_user_content</OFTPFileUserField>
      <RecordLength>n<RecordLength>
      <RecordDelimiter>one_or_two_delimiters<RecordDelimiter>

```

```

    </properties>
  </document>
    <MessageID>message_id<MessageID
    _OR_
    <PrimaryDocument SCIObjectID="document_id_1" />
  </document>
<FILE>
</DataItem>

```

Defining XML Node Name Parameters

To define the XML node name parameters:

1. Edit the PartnerProfile.xml file at *install_dir/install/properties* if you are using the default partner profile.
2. Make the following changes to the file.

The following table describes the parameters that need to match the definitions in the OFTP partner profile. Make sure that all required parameters belonging to your logical and physical contract are configured correctly, including all parameters of the logical and physical partners referenced in the contract part. Details are described in the default partner profile in XML comments.

Parameter	Description
LogicalPartnerContract	The unique name of the logical partner contract as defined in the Partner Profile XML file. This is an 80-character string. Required.
OFTPVirtualFilename	The OFTP Virtual File name. Defined according to the bilateral agreement with your trading partner. This is a 26-character string. Optional. If omitted, a default file name is taken from the partner profile contract. Note: The virtual file name, date, and time are used to uniquely define a file.
Date	The date tag used to send the message. This is a six-digit or eight-digit number. Format is as follows: <ul style="list-style-type: none"> ◆ YYMMDD (used for version 1.2 and version 1.3) ◆ YYYYMMDD (used for version 1.4) Optional. Note: The virtual file name, date, and time are used to uniquely define a file.
Time	The time stamp from when a file is made available for transmission at the sender's location. This is a six-digit or ten-digit number. Format is as follows: <ul style="list-style-type: none"> ◆ HHMMSS (used for version 1.2 and version 1.3) ◆ HHMMSS<four-digit counter> (used for version 1.4) Optional, but it is recommended that you specify the date and time stamp when the application created the file. Note: The virtual file name, date, and time are used to uniquely define a file.

Parameter	Description
FileFormat	This field specifies the format of the virtual file. Valid values are: U - unstructured binary file T - text file F - fixed-length record binary file V - variable-length record binary file Optional. If specified, overrides value in the partner profile.
OFTPFileUserField	Used as defined by your bilateral agreement with your partner. Optional. If specified, overrides value in the partner profile.
RecordDelimiter	One or two characters used to separate records if the FileFormat = V. This is a four-character string, with characters separated by commas or spaces. Optional. If specified, overrides value in the partner profile. Examples: <ul style="list-style-type: none"> ◆ 10 = line feed ◆ 13, 10 = carriage return and line feed
RecordLength	Number of bytes to be transferred in one OFTP record. Applies only to FileFormat = F. Optional. If specified, overrides value in the partner profile.
MessageID	The unique number identifying the message in the partner mailbox. Specify either MessageID or PrimaryDocument, not both.
PrimaryDocument	The Application document to send. Specify either MessageID or PrimaryDocument, not both.

XML Input Structure for Message Type EERP

The following is an example of the input structure for an EERP:

```
<DataItem>
  <EERP>
    <properties>
      <LogicalPartnerContract>log_partner_name</LogicalPartnerContract>
      <OFTPVirtualFilename>virtual_filename</OFTPVirtualFilename>
      <Date>date</Date>
      <Time>time</Time>
    </properties>
  </EERP>
</DataItem>
```

Parameter	Description
LogicalPartnerContract	The unique name of the logical partner contract as defined in the Partner Profile XML file. This is an 80-character string. Required.
OFTPVirtualFilename	The OFTP Virtual File name. Defined according to the bilateral agreement with your trading partner. This is a 26-character string. Optional. If omitted a default file name is taken from the partner profile contract. Note: The virtual file name, date, and time are used to uniquely define a file.

Parameter	Description
Date	<p>The date tag used to send the message. This is a six-digit or eight-digit number. Format is as follows:</p> <ul style="list-style-type: none"> ◆ YYMMDD (used for version 1.2 and version 1.3) ◆ YYYYMMDD (used for version 1.4) <p>Optional.</p> <p>Note: The virtual file name, date, and time are used to uniquely define a file.</p>
Time	<p>The time stamp from when a file is made available for transmission at the sender's location. This is a six-digit or ten-digit number. Format is as follows:</p> <ul style="list-style-type: none"> ◆ HHMMSS (used for version 1.2 and version 1.3) ◆ HHMMSS<four-digit counter> (used for version 1.4) <p>Optional, but it is recommended that you specify the date and time stamp when the application created the file.</p> <p>Note: The virtual file name, date, and time are used to uniquely define a file.</p>

XML Input Structure for Message Type NERP

The following is an example of the input structure for an NERP:

```

<DataItem>
  <NERP>
    <properties>
      <LogicalPartnerContract>log_partner_name</LogicalPartnerContract>
      <Creator>creator_of_nerp</Creator>
      <OFTPVirtualFilename>virtual_filename</OFTPVirtualFilename>
      <Date>date</Date>
      <Time>time</Time>
      <Reason>reasoncode</Reason>
    </properties>
  </NERP>
</DataItem>

```

Parameter	Description
LogicalPartnerContract	The unique name of the logical partner contract as defined in the Partner Profile XML file. This is an 80-character string. Required.
Creator	Creator of the NERP. This is a 25-character string. Required.
OFTPVirtualFilename	The OFTP Virtual File name. Defined according to the bilateral agreement with your trading partner. This is a 26-character string. Required. Note: The virtual file name, date, and time are used to uniquely define a file.
Date	<p>The date tag used to send the message. Required. This is a six-digit or eight-digit number. Format is as follows:</p> <ul style="list-style-type: none"> ◆ YYMMDD (used for version 1.2 and version 1.3) ◆ YYYYMMDD (used for version 1.4) <p>Note: The virtual file name, date, and time are used to uniquely define a file.</p>

Parameter	Description
Time	<p>The time stamp from when a file is made available for transmission at the sender's location. Required. This is a six-digit or ten-digit number. Format is as follows:</p> <ul style="list-style-type: none"> ◆ HHMMSS (used for version 1.2 and version 1.3) ◆ HHMMSS<four-digit counter> (used for version 1.4) <p>Note: The virtual file name, date, and time are used to uniquely define a file.</p>
Reason	Reason code for the NERP. Required. Valid values are two-digit numbers defined in the OFTP specification.

Partner Mailboxes

To use OFTP with the mailbox system, you must have a standard sub-mailbox layout for storing scheduled and received OFTP files in your Mailbox Virtual Root. Files to be sent are stored in the sub-mailbox named Inbox. Received files are stored in the Virtual Root of the Mailbox User.

Error Messages

The following error messages will be only displayed in the Status Report for the instance of the OdetteFTP Queue Handler service.

Advanced Status	Description
OFTP_INITIALIZATION_FAILED	An error occurred when initializing required components used in the Service. For example, the Service Framework providing logging services did not start successfully or the configuration of the Partner Profile contained errors. See OdetteFTP.log for details.
QUEUE_DATAITEM_FAILED	Either the Input XML structure is invalid (syntax errors, missing fields, invalid field length) or errors occurred when storing the entry in the process queue (table OFTP Object). See OdetteFTP.log for details.
OdetteFTPQueueHandler_FAILURE	A general error occurred during processing that does not fit into the categories listed above. The reason is noted in the status report/OdetteFTP.log.

OdetteFTP Scheduler Service

The following table provides an overview of the OdetteFTP Scheduler service:

System name	OdetteFTP Scheduler Service
Graphical Process Modeler (GPM) category	All Services
Description	The OdetteFTP Scheduler is used to initiate OFTP sessions for sending mailbox messages based on a time schedule.
Business usage	The scheduler starts a business process, configured in the Partner Profile Time Schedule, which calls the OdetteFTP adapter. The OdetteFTP Scheduler service initiates the OdetteFTP adapter (if the partner contract defines Mailbox Mode) based on a partner specific schedule.
Usage example	<p>The OdetteFTP Scheduler is used to initiate OFTP sessions for sending mailbox messages based on a time schedule. A time schedule and a business process name may be defined in the Partner Profile for each Physical Partner Contract. When the next scheduled point of time is reached following actions are performed depending on the action type of the scheduled entry:</p> <ul style="list-style-type: none">◆ Action Type = ConditionalCall: The OdetteFTP Scheduler looks up the mailbox related to the Physical Partner Contract and checks for files to send (Status = Scheduled or Retry). If yes, it initiates a business process and OFTP session.◆ Action Type = UnconditionalCall: The OdetteFTP Scheduler triggers a business process and passes the PhysicalPartnerContract name to it. The business process calls an OdetteFTP instance to look up the Mailbox related to the Physical Partner Contract. An OFTP session sends the mailbox messages to the Remote Physical Partner. Use this action type to poll a partner mailbox without sending messages to the partner.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported platforms for Application
Related services	OdetteFTP adapter (Mailbox mode only) OdetteFTP Queue Handler service
Application requirements	The OdetteFTP Scheduler service is used in conjunction with the OdetteFTP adapter in Mailbox mode.
Initiates business processes?	Yes
Invocation	No
Business process context considerations	None
Returned status values	None

Restrictions	None
Persistence level	System default
Testing considerations	Not applicable

Implementing the OdetteFTP Scheduler Service

The OdetteFTP Scheduler is not called by a business process. It is started automatically when Application is started and shuts down when Application is stopped. Enable and disable the OdetteFTP Scheduler service by starting and stopping the service in the Administration menu.

Configuring the OdetteFTP Scheduler Service

To configure the OdetteFTP Scheduler adapter, you must specify field settings in Application or the GPM:

Field	Description
CleanupInterval	Length, in hours, how long records that did not reach a terminated status should remain in the process history. Valid value is any integer. Required.

Partner Mailboxes

To use OFTP with the mailbox system, you must have a standard sub-mailbox layout for storing scheduled and received OFTP files in your Mailbox Virtual Root. Files to be sent are stored in the sub-mailbox named Inbox. Received files are stored in the Virtual Root of the Mailbox User.

Oftp Adapter

The following table provides an overview of the Oftp adapter:

Note: See the *OdetteFTP adapter* for the most current information on using OFTP with Application.

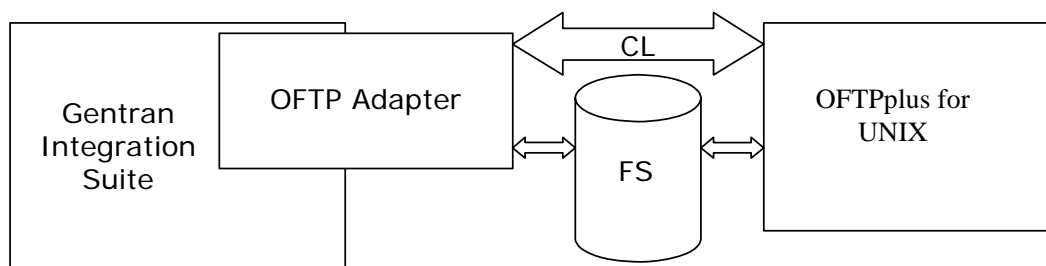
System name	Oftp Adapter
Graphical Process Modeler (GPM) category	None
Description	This adapter is used to exchange files with trading partners via the Odette FTP (Oftp) protocol.
Business usage	The Oftp Adapter uses OFTPplus for UNIX to support point-to-point communication between trading partners using an X.25, X.28(PAD), X.31(ISDN) or TCP/IP connection. It provides a file transfer service to communicate between peers. Most major European automobile manufacturers and their suppliers now use Oftp. It is also used by the chemical industry, white goods manufacturers, and is currently being adopted by other sectors such as banking and transportation.
Usage example	Trading Partner A requests a session with Trading Partner B. Trading Partner A sends the Start Session Ready Message command, indicating that the application is communicating via Oftp. Trading Partner A requests permission to send a file. The request contains information such as the origin and destination of the file, its name and physical size. Then Trading Partner A sends the actual file to Trading Partner B. Immediately after the file has been transferred, Trading Partner A sends the End of File Identification command, which contains control totals to ensure the integrity of the sent file. When the file reaches its ultimate destination, Trading Partner B returns an End-to-End Response command to Trading Partner A to inform them that the file has been received. Trading partner B then issues an End of Session Identification (ESID) command to request that the communication session be terminated. Trading Partner A terminates the session.
Preconfigured?	No
Requires third party files?	The Oftp adapter requires the installation of OFTPplus. This is the Oftp part of the Gentran:Server for UNIX product and is available from Sterling Commerce.
Platform availability	<ul style="list-style-type: none">◆ Sun Solaris◆ HP-UX◆ IBM-AIX
Related services	None
Application requirements	An Oftp server must exist at the external trading partner location. Information about this server must be configured in the trading profile.
Initiates business processes?	In the Inbound case (if an Oftp file is received) a business process is started.
Invocation	This adapter can be initiated from a business process. It can also be initiated by inbound communication from a trading partner.

Business process context considerations	The Oftp adapter takes the Primary Document and potential relevant extra BPML Parameters from the WFC
Returned status values	<ul style="list-style-type: none"> ◆ 0 – Success ◆ 1 – Error
Restrictions	You must have a valid installation of the OFTPplus for UNIX software. If you use the waiting mode, depending on the Network Infrastructure of your Oftp Communication Partner (e.g. a VAN like IBM or GE) the Oftp adapter could block the business process for a very long time (hours to days).
Persistence level	System default
Testing considerations	You need an Installation of the OFTPplus for UNIX Software

Prerequisites

OFTPplus for UNIX must be purchased, installed and configured before you can use it with the Oftp adapter.

The working setup will look like this:



How the Oftp Adapter Works

The Oftp adapter makes the functionality of OFTPplus for UNIX available in Application. You can use the Oftp adapter to send documents in synchronous or asynchronous mode. In addition, it can be configured to wait for the EERP (End to End ResPonse) received from the Oftp partner acknowledging that the file has been received.

Oftp Adapter Modes

The following table describes the different modes of the Oftp adapter:

Mode	Wait	Behavior
Asynchronous	No	The Primary Document will be put into the OFTPplus transmit queue for a certain business partner, but transmission will not be triggered. The call to the adapter returns immediately.

Mode	Wait	Behavior
Asynchronous	Yes	The Primary Document will be put into the OFTPplus transmit queue for the partner, no transmission is triggered. But the adapter sets the business process into a Consume state. It will wait for an incoming EERP that is produced by the Partner on successful reception of the transferred file.
Synchronous	No	The Primary Document is put into the OFTPplus transmit queue and transmission is immediately started. After transmission has finished, the Adapter returns control to the business process.
Synchronous	Yes	The Primary Document is put in the OFTPplus transmit queue and transmission is immediately started. The adapter will bring the business process into Consume mode. It will wait for an incoming EERP that is produced by the partner on successful reception of the data file.

Sample Business Scenarios

1. You have many files to send each day, but to save costs, you want them to be transmitted once every evening in one batch. You set up a business process to call the Oftp adapter in Asynchronous mode and set Wait to either Yes or No depending on whether the business process should be halted until your trading partner acknowledges the files. You set up a business process scheduled to run every evening that calls the Oftp adapter in synchronous mode to start the transmission. The Oftp product will send all scheduled files and restart all waiting business processes.
2. You want to place just in time orders to a trading partner. You would set up your business process in synchronous mode to be sure that the order file gets transmitted immediately.

The Oftp adapter also gets a notice for every incoming file and can bootstrap new business processes for every incoming file.

The Oftp adapter can create Notification files in the Notification directory. There are two scripts shipped with the adapter that have to be used in the OFTPplus User-Exits. These scripts will be called for certain events:

- Successful reception of a file

- Processing of an incoming EERP

For each of those two events a Notification file is created that is just a simple XML structure.

The Oftp adapter will poll the Notification directory in intervals. Once it picks up a notification it will react to it. If it is a File Type notification, a new business process will be bootstrapped and the Transmission file will be put into the business as the Primary Document. The Notification file will contain a UNIXFN field. The Oftp adapter will use the content of that field to find the corresponding file.

If the Notification type was EERP, the Oftp adapter will search for a matching business process that is still waiting for this EERP in Consume mode. It will produce a message to restart the waiting business process. The unique key that is used for that is the UNIXFN field of the Notification.

For outbound Transmissions, the Oftp adapter calls the OFTPplus commands, Oftpsys and hOftp, directly on the command line. In case the Oftp adapter gets called and there is a Primary Document available, it is written into the Oftp Data Directory, then Oftpsys is called, which will schedule the file. If needed, hoftp is called subsequently to kick off the transmission.

If you want to send a file to a trading partner, create an instance of the Oftp adapter, configure all parameters and activate the adapter. You can then use the adapter instance by just passing a Primary Document to it. It will take the document, put it into the queue for the trading partner and eventually send it directly. If the waiting mode is active, the Oftp adapter will bring the business process into the consume mode and wait until a corresponding EERP is received.

Implementing the Oftp Adapter

To implement the Oftp adapter, complete the following tasks:

1. Activate your license for the Oftp adapter.
2. Create an Oftp adapter configuration. See *Creating a Service Configuration*.
3. Configure the adapter. See *Configuring the Oftp Adapter* on page 1037.
4. Create and enable a business process that includes the Oftp adapter.
5. Test the business process and the adapter.
6. Run the business process.


Configuring the Oftp Adapter

To configure the Oftp adapter, you must specify field settings in Application:

Note: The field names in parentheses represent the corresponding field names in the Graphical Process Modeler. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Using Service Groups</i>.</p>
OFTPPlus Installation Directory (OftpDir)	Directory where the OFTPplus software is installed. Required
Partner Profile (Profile)	Name of the partner profile that will be addressed by OFTPplus. This is the receiver of the file. Optional.
Run in Trace Mode (Tracemode)	Enables tracing in OFTPplus. Tracing information will be written into the OFTPplus directory.

Field	Description
Start business process (BusinessProcessName)	Name of the business process to start when files are received by OFTPplus. Optional. The default is Not Applicable.
Notification files directory (NotifyDir)	Directory where notification files are maintained. This is the directory that OFTPplus and the Oftp adapter use to exchange management information. Required.
Pick up notifications	Whether to pick up notifications. Optional.
Check for notifications every (seconds) (PollingPeriod)	The time interval (in seconds) when notifications should be checked in the Notification directory. Required. Valid values include any positive integer between 0 and 9999999999. The default value is 3600.
Integration Mode (Mode)	Directs handling of the file for processing. Required. Valid values: <ul style="list-style-type: none"> ◆ Put file in send queue without sending it – Asynchronous mode with no wait ◆ Put file in send queue and wait for response – Asynchronous mode with wait ◆ Send immediately without waiting for response – Synchronous mode with no wait ◆ Send immediately and wait for response – Synchronous mode with wait See <i>Oftp Adapter Modes</i> .
Max. time to wait for response (seconds, 0=unlimited) (Wait)	Maximum length of time (in seconds) to wait for the response from a finished transmission. This is the time that a business process will stay in consume mode. Required. Valid values include any positive integer between 0 and 9999999999. Specify 0 for unlimited wait time. The default value is 3600.
Send File Data Path (for oftpsys) (Path)	Specifies the interchange Directory. This is where the oftpsys application waits for the files that are being sent. Required.
Generate Virtual File Name automatically (AutoFilename)	Whether a virtual file name should be generated automatically. Optional. The default is not selected.
Oftp Virtual Filename Prefix (Filename)	Prefix for the Virtual Filename. This parameter is only required if Generate Virtual File Name automatically is not selected.
Oftp Virtual Filename Time Stamp format (FileDateFormat)	Time stamp format for the specified virtual file name. Optional. This parameter is only displayed if Generate Virtual File Name automatically is not selected. This parameter uses the Java SimpleDateFormat class. For more information, see http://java.sun.com .
Oftp Virtual Filename Suffix (FilenameSuffix)	Suffix for the Virtual Filename. Optional. This parameter is only displayed if Generate Virtual File Name automatically is not selected.

Field	Description
User	<p>User ID to use with the business process.</p> <p>Type the user ID, or click the  icon and select a user ID from the list.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Any valid Application user ID <p>Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>

Business Process Configuration

No Oftp adapter parameters are required to be configured with the GPM. However, there are two parameters available in the GPM configuration that allow for dynamic operation.

The following table describes the fields used to configure the Oftp adapter in the GPM:

Field	Description
CommunicationCommand	Overwrites the command name that is executed to start the communication. Optional.
SchedulingCommand	Overwrites the command name that is executed to schedule a file for transmission. Optional.

Parameters Passed during Incoming Oftp Transmission

In case a transmission is coming in via Oftp, the Oftp adapter will start a new business process, attach the transmission as a primary document, and additionally fill these fields:

Field	Description
INFTRIGGER/UNIX	UNIX Filename
INFTRIGGER/VIRTUAL	Virtual Filename provided by Partner
NFTRIGGER/SFIDDEST	Destination Address in SFID
INFTRIGGER/SFIDORIG	Originator Address in SFID
INFTRIGGER/PID	PID of the Starting Oftp process

Error Messages

The following error messages will only be displayed in the Status Report in this instance of the Oftp adapter:

Timeout while waiting for a EERP

Invalid Profile was selected
Communication Error

Example Business Process

The following sample BPML will pass a primary document to the Oftp adapter. The Oftp adapter will either queue the file, transmit the file, or do both depending on its configuration.

```
<process name="SendOFTPFile">
  <sequence name="out">
    <operation name="SendFile">
      <participant name="OFTP"/>
      <output message="msg">
        <assign to="." from="*"></assign>
      </output>
      <input message="msg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Oracle AQ JMS Queue Adapter

The following table provides an overview of the Oracle® AQ JMS Queue adapter:

System name	Oracle AQ JMS Queue Adapter
Graphical Process Modeler (GPM) category	All, Queuing
Description	Sends messages to and receives messages from a remote Oracle AQ JMS queue. Use this adapter when you want to exchange messages with a remote Oracle AQ JMS Queue server as part of a business process within Application.
Preconfigured?	No
Requires third party files?	Yes. aqapi.jar, classes12.zip
Platform availability	All supported Application platforms
Related services	Oracle AQ JMS Topic adapter
Application requirements	No
Initiates business processes?	Yes. See <i>How the Oracle AQ JMS Queue Adapter Works</i> on page 1041.
Invocation	Runs as part of a business process.

How the Oracle AQ JMS Queue Adapter Works

The following steps summarize how the Oracle AQ JMS Queue adapter works within a business process when you send or receive messages.

When Sending Messages to a Remote Oracle AQ JMS Queue

1. The Oracle AQ JMS Queue adapter connects to the remote Oracle AQ JMS Queue server and sends a message.
2. Application proceeds with the next step in the business process.

When Receiving Messages from a Remote Oracle AQ JMS Queue

1. The Oracle AQ JMS Queue adapter connects to the remote Oracle AQ JMS Queue server and checks for and receives any available data.
2. The Oracle AQ JMS Queue adapter initiates the appropriate business process (the business process you specified when you configured the adapter).
3. Application proceeds with the next step in the business process.

Note: The Oracle AQ API does not allow for a timeout. The Oracle AQ adapter may have threads open waiting on a response from the Oracle AQ Server for an unknown time period.

If this occurs and if you see open Oracle AQ processes in Application not responding for a long time, break the connections on the AQ server to force AQ to close connections with Application and release the threads. If this is a consistent problem in your Oracle environment, review the Application Performance and Tuning documentation for instructions on how to manage thread resources and queues.

Implementing the Oracle AQ JMS Queue Adapter

To implement the Oracle AQ JMS Queue adapter, complete the following tasks:

1. Activate your license for the Oracle AQ JMS Queue adapter. For information, see *An Overview of Implementing Services*.
2. Verify that you have Application and Oracle installed.
3. If it is running, stop Application.
4. From the bin directory where Application is installed, install the necessary third party files (aqapi.jar and classes12.zip) using the install3rdParty script included with Application. Use the Oracle version number and the absolute path to the files to enter commands similar to the following examples:
 - ◆ On a UNIX or Linux system, enter:

```
./install3rdParty.sh Oracle 10_1 -d absolutePath  
absolutePath/classes12.zip
```
 - ◆ On a Windows system, enter:

```
install3rdParty.cmd Oracle 10_1 -d absolutePath/aqapi.jar  
install3rdParty.cmd Oracle 10_1 -d absolutePath/classes12.zip
```
5. Restart Application.
6. Set up a queue in Oracle AQ. For information, see your Oracle AQ documentation.
7. Create an Oracle AQ JMS Queue adapter configuration. For information, see *Managing Services and Adapters*.
8. Configure the Oracle AQ JMS Queue adapter. For information, see *Configuring the Oracle AQ JMS Queue Adapter* on page 1042.
9. Use the Oracle AQ JMS Queue adapter in a business process.

Configuring the Oracle AQ JMS Queue Adapter

To configure the Oracle AQ JMS Queue Adapter, you must specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.

Field	Description
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None –You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
System ID (sid)	Name of the Oracle AQ system.
Host (host)	Host address of the Oracle AQ system.
Port (port)	Port number of the Oracle AQ system.
Driver (driver)	Driver name used to access the Oracle AQ system.
User Name (userName)	User name used to access the Oracle AQ system.
Password (passWord)	Password associated with the user name used to access the Oracle AQ system.
Schema Name (queueSchema)	Schema name associated with the queue.
Queue Name (queueName)	Name of the queue to access.
Queue Type (Action)	<p>Type of queue to access. Valid values are:</p> <ul style="list-style-type: none"> ◆ Queue Send – Send messages. ◆ Queue Receive Sync – Receive messages.
Message Type (payload)	<p>Type of message to send. Valid values are:</p> <ul style="list-style-type: none"> ◆ Bytes Message – Send a message containing a stream of uninterpreted bytes. ◆ Object Message – Send a message containing a serializable Java object. ◆ Stream Message – Send a message containing a stream of Java primitives. ◆ Text Message – Send a message containing a Java String type.
Buffer Size	Size of the buffer when receiving data. Enables you to fine-tune the performance of the adapter according to data expectations.

Field	Description
Filename (filename)	File name to assign, if any, and the file name extension. A unique file name generator placeholder, %^, can be used to generate a sequence in the form <i>yyyymmddhhmmsslll</i> .
Business process (initialWorkflowId)	Business process you want the Oracle AQ JMS Queue adapter to start, if any. Required only if the Oracle AQ JMS Queue adapter starts a business process.
Do not use schedule	If field is selected, this service does not start a business process and does not run on a schedule.
Run service based on timer every	Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup.
Run service daily at	Valid values are the hour and minutes at which to run the service, daily. Indicate whether you want the service to run at startup.
Run service weekly on	Valid values are the day of the week, the hour, and the minutes at which to run the service. Indicate whether you want the service to run at startup.

Oracle AQ JMS Topic Adapter

The following table provides an overview of the Oracle[®] AQ JMS Topic adapter:

System name	Oracle AQ JMS Topic Adapter
Graphical Process Modeler (GPM) category	All, queuing
Description	Sends messages to and receives messages from a remote Oracle AQ JMS topic. Use this adapter when you want to exchange messages with a remote Oracle AQ JMS topic server as part of a business process within Application.
Preconfigured?	No
Requires third party files?	Yes. aqapi.jar, classes12.zip
Platform availability	All supported Application platforms
Related services	Oracle AQ JMS Queue adapter
Application requirements	No
Initiates business processes?	Yes. See <i>How the Oracle AQ JMS Topic Adapter Works</i> on page 1045.
Invocation	Runs as part of a business process.

How the Oracle AQ JMS Topic Adapter Works

The following steps summarize how the Oracle AQ JMS Topic adapter works within a business process when you send or receive messages.

When Sending Messages to a Remote Oracle AQ JMS Topic

1. The Oracle AQ JMS Topic adapter connects to the remote Oracle AQ JMS topic server and sends a message.
2. Application proceeds with the next step in the business process.

When Receiving Messages from a Remote Oracle AQ JMS Topic

1. The Oracle AQ JMS Topic adapter connects to the remote Oracle AQ JMS Topic server and checks for and receives any available data.
2. The Oracle AQ JMS Topic adapter initiates the appropriate business process (the business process you specified when you configured the adapter).
3. Application proceeds with the next step in the business process.

Implementing the Oracle AQ JMS Topic Adapter

To implement the Oracle AQ JMS Topic adapter, complete the following tasks:

1. Activate your license for the Oracle AQ JMS Topic adapter. For information, see *An Overview of Implementing Services*.
2. Verify that you have Application and Oracle installed.
3. If it is running, stop Application.
4. From the bin directory where Application is installed, install the necessary third party files (aqapi.jar and classes12.zip) using the install3rdParty script included with Application. Use the Oracle version number and the absolute path to the files to enter commands similar to the following examples:
 - ◆ On a UNIX or Linux system, enter:

```
./install3rdParty.sh Oracle 10_1 -d absolutePath/aqapi.jar
./install3rdParty.sh Oracle 10_1 -d absolutePath/classes12.zip
```
 - ◆ On a Windows system, enter:

```
install3rdParty.cmd Oracle 10_1 -d absolutePath/aqapi.jar
install3rdParty.cmd Oracle 10_1 -d absolutePath/classes12.zip
```
5. Restart Application.
6. Set up a topic in Oracle AQ. For information, see your Oracle AQ documentation.
7. Create an Oracle AQ JMS Topic adapter configuration. For information, see *Managing Services and Adapters*.
8. Configure the Oracle AQ JMS Topic adapter. For information, see *Oracle AQ JMS Topic Adapter* on page 1045.
9. Use the Oracle AQ JMS Topic adapter in a business process.

Configuring the Oracle AQ JMS Topic Adapter

To configure the Oracle AQ JMS Topic adapter, you must specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
System ID (sid)	Name of the Oracle AQ JMS topic system.
Host (host)	Host address of the Oracle AQ JMS topic system.
Port (port)	Port number of the Oracle AQ JMS topic system.
Driver (driver)	Driver name used to access the Oracle AQ JMS topic system.
User Name (userName)	User name used to access the Oracle AQ JMS topic system.
Password (passWord)	Password associated with the user name used to access the Oracle AQ JMS topic system.
Schema Name (topicSchema)	Schema name associated with the topic.
Topic Name (topicName)	Name of the topic to access.
Topic Type (Action)	<p>Type of topic to access. Valid values are:</p> <ul style="list-style-type: none"> ◆ Topic Send – Send messages. ◆ Topic Receive Sync – Receive messages.
Message Type (payload)	<p>Type of message to send. Valid values are:</p> <ul style="list-style-type: none"> ◆ Bytes Message – Send a message containing a stream of uninterpreted bytes. ◆ Object Message – Send a message containing a serializable Java object. ◆ Stream Message – Send a message containing a stream of Java primitives. ◆ Text Message – Send a message containing a Java String type.
Buffer Size	Size of the buffer when receiving data. Enables you to fine-tune the performance of the adapter according to data expectations.

Field	Description
Filename (filename)	File name to assign, if any, and the file name extension. A unique file name generator placeholder, %^, can be used to generate a sequence in the form <i>yyyymmddhhmmsslll</i> .
Business process (initialWorkflowId)	Business process you want the Oracle AQ JMS Topic adapter to start, if any. Required only if the Oracle AQ JMS Topic adapter starts a business process.
Do not use schedule	If this field is selected, this adapter does not start a business process and does not run on a schedule.
Run service based on timer every	Valid values are the hour and minutes at which to run the adapter. Indicate whether you want the adapter to run at startup.
Run service daily at	Valid values are the hour and minutes at which to run the adapter, daily. Indicate whether you want the adapter to run at startup.
Run service weekly on	Valid values are the day of the week, the hour, and the minutes at which to run the adapter. Indicate whether you want the adapter to run at startup.

Oracle E-Business Suite Adapter Configuration Service

The following table provides an overview of the Oracle® E-Business Suite Adapter Configuration service:

System name	OracleEBusinessConfig
Graphical Process Modeler (GPM) categories	All Services, Applications > ERP
Description	The Oracle E-Business Suite Adapter Configuration service is a custom service developed for the Oracle E-Business Suite Adapter. It is used to set configuration parameters for the adapter BPML. It does this by populating process data with values specified in the service configuration.
Business usage	The Oracle E-Business Suite adapter is BPML-based. The Oracle E-Business Suite Adapter Configuration service enables you to configure the adapter using Application.
Usage example	If you want to use the Oracle E-Business Suite adapter, you specify configuration parameters using this service. The BPML below provides a usage example: <pre><operation name="Config"> <participant name="OracleEBusinessAdapterConfig"/> <output message="Xout"> <assign to="." from="*" /> </output> <input message="Xin"> <assign to="." from="*" /> </input> </operation></pre>
Preconfigured?	No. An instance named OracleEBusinessAdapterConfig is installed but must be configured. OracleEBusinessAdapterConfig is the default configuration used by the OracleEBusinessReceive business process.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Oracle E-Business Suite adapter (BPML-based adapter), Oracle E-Business Suite Message service
Application requirements	None
Initiates business processes?	No
Invocation	This adapter disregards parameters sent from the BPML in the output message, so the configuration parameters cannot be overridden with the GPM. The service configuration parameter values are used to populate process data (via the input message to the BPML).
Business process context considerations	See Invocation.

Returned status values	<ul style="list-style-type: none"> ◆ Success – The service successfully populated process data with the configuration parameters. ◆ Error – A system error occurred.
Restrictions	None. Any number of configurations can be created, but typically a single configuration can be used by any number of business processes.
Persistence level	Default (Full)
Testing considerations	To test it, create a configuration, then run it from a business process as shown in the example business process. Verify that process data contains all the parameters listed in the <i>Parameters Passed from Service to Business Process</i> section.

Implementing the Oracle E-Business Suite Adapter Configuration Service

To implement the Oracle E-Business Suite Adapter Configuration service for use in a business process:

1. Create an Oracle E-Business Suite Adapter Configuration service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Oracle E-Business Suite Adapter Configuration service. For information, see *Configuring the Oracle E-Business Suite Adapter Configuration Service* on page 1050.
3. Use the Oracle E-Business Suite Adapter Configuration service in a business process.

Configuring the Oracle E-Business Suite Adapter Configuration Service

This service uses only the parameter values configured for the service configuration through the Application Administration console. It disregards any parameters that are set using the GPM (in the output message) or in the BPML. Therefore, the service configuration parameters cannot be overridden using the GPM or BPML.

The Output Element column shows the element name populated in process data by the service at run time. They are output parameters only. Notice that all elements are contained within the parent element OracleEBusiness.

To configure the Oracle E-Business Suite Adapter Configuration service, you must specify settings for the following fields in Application:

Field Name	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field Name	Description
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
The Trading Partner Contract for sending	Contract used for sending messages to Oracle e-Business suite. Required. Valid value is any valid trading partner contract ID.
Insert Date and Time into OAG doc at run time	Insert Date and Time into OAG doc CNTROLAREA at run time. Required. Valid values are true and false.
Timeout (minutes) to wait for Confirm BOD	Number of minutes to wait for a Confirm BOD response from Oracle. Required. Valid value is any non-negative integer.
Business Process to handle received messages	The name of business process to handle incoming messages from Oracle. If not specified, a business process named OracleEBusinessRecv_<VERB>_<NOUN> will handle incoming messages, where <VERB> and <NOUN> are taken from the CNTROLAREA of the incoming message. Optional. Valid value is any valid business process name.
Business Process to handle unknown Confirm BODs	Name of the business process that runs if a Confirm BOD is received and no matching request can be found. Optional. Valid value is any valid business process name.
Automatically send Confirm BOD for received messages?	Determines if the adapter automatically sends a Confirm BOD when required. Required. Valid values are true and false. If false, you are responsible for sending a confirm BOD, if required.
Name of Confirm BOD template file	Name of file containing the Confirm BOD template. Required if value in Automatically send Confirm BOD for received messages field is true. Valid value is any valid file name.
Directory of Confirm BOD template file	The directory containing the Confirm BOD template file. Required if value in Automatically send Confirm BOD for received messages field is true. Valid value is any valid directory name.
Request Type	The OTA protocol REQUEST_TYPE parameter. The default value SEND should be used for normal transactions. Required. Valid values are: <ul style="list-style-type: none"> ◆ SEND (default) ◆ AUTH ◆ AUTH2 ◆ EME For information about these values, see your Oracle documentation.
Protocol	The OTA protocol PROTOCOL_TYPE parameter. Required. Valid values are HTTP and HTTPS.
Oracle EBusiness Suite user name	Valid user name for the Oracle E-Business Suite. Required. Valid value is any valid username for Oracle E-Business Suite.

Field Name	Description
Oracle EBusiness Suite password	Valid password for the selected user name. Required. Valid value is a valid password for the selected user name.

Parameters Passed from Business Process to Service

No parameters are passed from the business process to the service; the service will ignore any parameters that are passed to it from a business process.

Output from Service to Business Process

The following table contains the parameters passed from the Oracle E-Business Suite Adapter Configuration service to a business process:

Note: The output parameters are all contained within the parent element OracleEBusiness.

Field Name	Description
The Trading Partner Contract for sending (OracleEBusiness/ SendingContractID)	Contract used for sending messages to Oracle e-Business suite. Required. Valid value is any valid trading partner contract ID.
Insert Date and Time into OAG doc at run time (OracleEBusiness/ InsertDateTime)	Insert Date and Time into OAG doc CNTROLAREA at run time. Required. Valid values are true and false.
Timeout (minutes) to wait for Confirm BOD (OracleEBusiness/ ConfirmBodTimeoutMinutes)	Number of minutes to wait for a Confirm BOD response from Oracle. Required. Valid value is any non-negative integer.
Business Process to handle received messages (OracleEBusiness/ ReceiveMessageBPName)	The name of business process to handle incoming messages from Oracle. If not specified, a business process named OracleEBusinessRecv_<VERB>_<NOUN> will handle incoming messages, where <VERB> and <NOUN> are taken from the CNTROLAREA of the incoming message. Optional. Valid value is any valid business process name.
Business Process to handle unknown Confirm BODs (OracleEBusiness/ UnknownConfirmBodBPName)	Name of the business process that runs if a Confirm BOD is received and no matching request can be found. Optional. Valid value is any valid business process name.
Automatically send Confirm BOD for received messages? (OracleEBusiness/ AutoSendConfirmBod)	Determines if the adapter automatically sends a Confirm BOD when required. Valid values are true and false. If false, you are responsible for sending a confirm BOD, if required. Required. Valid values are true and false.
Name of Confirm BOD template file (OracleEBusiness/ ConfirmBodTemplateFileName)	Name of file containing the Confirm BOD template. Required if value in Automatically send Confirm BOD for received messages field is true. Valid value is any valid file name.

Field Name	Description
Directory of Confirm BOD template file (OracleEBusiness/ ConfirmBodTemplateDirectory)	The directory containing the Confirm BOD template file. Required if value in Automatically send Confirm BOD for received messages field is true. Valid value is any valid directory name.
Request Type (OracleEBusiness/ REQUEST_TYPE)	The OTA protocol REQUEST_TYPE parameter. The default value SEND should be used for normal transactions. Default is SEND. Required. Valid values are: <ul style="list-style-type: none"> ◆ SEND ◆ AUTH ◆ AUTH2 ◆ EME For more information about these values, see your Oracle documentation.
Protocol (OracleEBusiness/ PROTOCOL_TYPE)	The OTA protocol PROTOCOL_TYPE parameter. Required. Valid values are HTTP and HTTPS.
Oracle EBusiness Suite user name (OracleEBusiness/ USERNAME)	Valid user name for the Oracle E-Business Suite. Required. Valid value is any valid username for Oracle E-Business Suite.
Oracle EBusiness Suite password (OracleEBusiness/ PASSWORD)	Valid password for the selected user name. The password will be in an obscured format rather than clear text. The obscured password is required as input to the Oracle E-Business Suite Message Service. Required. Valid value is a valid password for the selected user name.

Oracle E-Business Suite Message Service

The following table provides an overview of the Oracle® E-Business Suite Message service:

System name	OracleEBusinessMessageService
Graphical Process Modeler (GPM) categories	All Services, Applications > ERP
Description	The Oracle E-Business Suite Message service is a custom service developed for the Oracle E-Business adapter. It is used to encode messages in the HTTP POST format expected by Oracle, and to decode HTTP POST messages received from Oracle.
Business usage	Oracle XML Gateway requires parameters in addition to the XML document being transmitted. This service gathers all of the parameters and constructs a message in the format Oracle expects (HTTP POST). For receiving messages, it parses the incoming HTTP POST message from Oracle to extract the XML payload and all the other parameters.
Usage example	<p>A user wants to use the Oracle E-Business adapter to send a purchase order (PO) to Oracle. The user defines all the Oracle parameters using the Oracle E-Business Suite Adapter Configuration service, and places the PO into the primary document. This service constructs a message in the format required by Oracle.</p> <p>Oracle sends an invoice to Application. The Oracle E-Business Suite Message service is used to parse the message, putting the invoice XML into the primary document and storing all other message parameters in process data.</p>
Preconfigured?	No
Requires third party files?	OAG DTDs must be checked in to Application for each document type exchanged with Oracle.
Platform availability	All supported Application platforms
Related services	Oracle E-Business Suite adapter (BPML-based adapter), Oracle E-Business Suite Adapter Configuration service
Application requirements	The user must have a valid username and password for the Oracle E-Business Suite
Initiates business processes?	None
Invocation	Runs by Oracle E-Business Suite business processes.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success: The service successfully encoded or decoded the message◆ Error: The service was unable to encode or decode the message.
Restrictions	None. Any number of configurations can be created, but typically a single configuration of the service is used by a number of business processes.
Persistence level	System default (Full)

Testing considerations	To test it, create an configuration, then run it from a business process as shown in the business process examples.
------------------------	---

Implementing the Oracle E-Business Suite Message Service

To implement the Oracle E-Business Suite Message service, complete the following tasks:

1. Create an Oracle E-Business Suite Message service configuration. For information, see *Managing Services and Adapters*.
2. Configure the Oracle E-Business Suite Message service. For information, see *Configuring the Oracle E-Business Suite Message Service* on page 1055.
3. Use the Oracle E-Business Suite Message service in a business process.

Configuring the Oracle E-Business Suite Message Service

To configure the Oracle E-Business Suite Message service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the adapter configuration.
ATTRIBUTE1	OTA parameter. Optional.
ATTRIBUTE2	OTA parameter. Optional.
ATTRIBUTE3	OTA parameter. Optional.
ATTRIBUTE4	OTA parameter. Optional.
ATTRIBUTE5	OTA parameter. Optional.
DOCUMENT_NUMBER	OTA parameter. Required if MODE is ENCODE.
MESSAGE_STANDARD	OTA parameter. Required if MODE is ENCODE. Valid value and default are OAG.
MESSAGE_TYPE	OTA parameter. Value must be XML with version 11.5.8 of XML gateway. Required if MODE is ENCODE.
MODE	Valid values are ENCODE (for sending a message to Oracle) and DECODE (for receiving a message from Oracle). Required.
PARTY_SITE_ID	OTA parameter PARTY_SITE_ID. Should match the Source Location Code in E-Business Suite trading partner definition. Required if MODE is ENCODE.
PARTYID	OTA parameter. Optional.

Field	Description
PASSWORD	OTA parameter. Valid password for the selected user name in obscured format (not clear text). The Oracle E-Business Suite Adapter Configuration service stores the password in the obscured format in the element OracleEBusiness/PASSWORD. Required if MODE is ENCODE. Valid value is a valid password for the selected user name (in obscured format).
PROTOCOL_ADDRESS	OTA parameter. Optional.
PROTOCOL_TYPE	OTA parameter. Required if MODE is ENCODE. Valid values are HTTP and HTTPS.
REQUEST_TYPE	OTA parameter REQUEST_TYPE. Valid values are SEND, AUTH, AUTH2, and EME. Default value is SEND. Required if mode is ENCODE.
TRANSACTION_SUBTYPE	OTA parameter TRANSACTION_SUBTYPE. Should match External Transaction SubType in E-Business Suite Transaction setup. Required if MODE is ENCODE.
TRANSACTION_TYPE	OTA protocol TRANSACTION_TYPE. Should match External Transaction Type in E-Business Suite Transaction setup. Required if MODE is ENCODE.
TRANSPORT_PROTOCOL	OTA parameter. Value must be OXTA with version 11.5.8 of XML gateway. Required if MODE is ENCODE.
TRANSPORT_PROTOCOL_VERSION	OTA parameter. Value must be 1.0 with version 11.5.8 of XML gateway. Required if MODE is ENCODE.
USERNAME	OTA parameter. Required if MODE is ENCODE. Valid value is any valid user name for Oracle E-Business Suite.

Parameters Passed from Service to Business Process

In ENCODE mode, the service will output a single parameter: PrimaryDocument. In DECODE mode, the service will output PrimaryDocument, and any other OTA parameters in the message (except PASSWORD) will be output within the parent element OracleEBusiness/Recv_OTA_Params. The OTA parameters will be output *only* if they are present in the message.

The following table describes the parameters passed from the service to a business process:

PrimaryDocument:
◆ ENCODE mode: an HTTP POST message to send to Oracle.
◆ DECODE mode: The XML payload from the incoming Oracle message.
OTA Parameters:
OracleEBusiness/Recv_OTA_Params/ REQUEST_TYPE
OracleEBusiness/Recv_OTA_Params/ TRANSACTION_TYPE
OracleEBusiness/Recv_OTA_Params/ TRANSACTION_SUBTYPE
OracleEBusiness/Recv_OTA_Params/ DOCUMENT_NUMBER

OracleEBusiness/Recv_OTA_Params/ PARTYID

OracleEBusiness/Recv_OTA_Params/ PARTY_SITE_ID

OracleEBusiness/Recv_OTA_Params/ PROTOCOL_TYPE

OracleEBusiness/Recv_OTA_Params/ USERNAME

OracleEBusiness/Recv_OTA_Params/ TRANSPORT_PROTOCOL

OracleEBusiness/Recv_OTA_Params/ TRANSPORT_PROTOCOL_VERSION

OracleEBusiness/Recv_OTA_Params/ MESSAGE_TYPE

OracleEBusiness/Recv_OTA_Params/ MESSAGE_STANDARD

OracleEBusiness/Recv_OTA_Params/ PROTOCOL_ADDRESS

OracleEBusiness/Recv_OTA_Params/ ATTRIBUTE1

OracleEBusiness/Recv_OTA_Params/ ATTRIBUTE2

OracleEBusiness/Recv_OTA_Params/ ATTRIBUTE3

OracleEBusiness/Recv_OTA_Params/ ATTRIBUTE4

OracleEBusiness/Recv_OTA_Params/ ATTRIBUTE5

Business Process Examples

The following examples illustrate using the Oracle E-Business Suite Message service in ENCODE and DECODE modes:

ENCODE Mode

```

<operation name="encode message">
  <participant name="OracleEBusinessMessageService" />
  <output message="outmsg">
    <assign to="OTAMessageServiceMode">ENCODE</assign>
    <assign to="REQUEST_TYPE" from="OracleEBusiness/REQUEST_TYPE/text()"/>
    <assign to="TRANSACTION_TYPE" from="OracleEBusiness/Noun/text()"/>
    <assign to="TRANSACTION_SUBTYPE" from="OracleEBusiness/Verb/text()"/>
    <assign to="DOCUMENT_NUMBER" from="OracleEBusiness/REFERENCEID/text()"/>
    <assign to="PARTY_SITE_ID"
      from="DocToDOM(PrimaryDocument)//CNTROLAREA/SENDER/LOGICALID/text()"/>
    <assign to="PROTOCOL_TYPE" from="OracleEBusiness/PROTOCOL_TYPE/text()"/>
    <assign to="USERNAME" from="OracleEBusiness/USERNAME/text()"/>
    <assign to="PASSWORD" from="OracleEBusiness/PASSWORD/text()"/>
    <assign to="ObscurePassword" from="OracleEBusiness/ObscurePassword/text()"/>
    <assign to="TRANSPORT_PROTOCOL">OXTA</assign>
    <assign to="TRANSPORT_PROTOCOL_VERSION">1.0</assign>
    <assign to="MESSAGE_TYPE">XML</assign>
    <assign to="MESSAGE_STANDARD">OAG</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>

```

DECODE Mode

```
<operation name="decode message">
  <participant name="OracleEBusinessMessageService"/>
  <output message="outmsg">
    <assign to="OTAMessageServiceMode">DECODE</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
```

Oracle E-Business Suite Adapter

The following table provides an overview of the Oracle® E-Business Suite adapter:

System name	None (BPML-based adapter and does not have a system name)
Graphical Process Modeler (GPM) category	None
Description	<p>The Oracle E-Business Suite adapter enables you to exchange messages with the Oracle E-Business Suite using the Oracle XML Gateway with HTTP or HTTPS. The adapter supports version 12.0.0 of the E-Business Suite.</p> <p>The adapter supports XML messages that meet the Open Applications Group (OAG) standard. See http://www.openapplications.org and the Oracle E-Business Suite documentation for more information.</p>
Business usage	The Oracle E-Business Suite adapter enables you to exchange messages with the Oracle E-Business Suite, which includes modules for financials, purchasing, accounts payable, and accounts receivable, among others.
Usage example	You want to send a purchase order to a trading partner Oracle E-Business Suite server. Build a business process to construct an OAG-format purchase order (PROCESS PO) and use the Oracle E-Business Suite adapter to send it to the trading partner. The trading partner may send back an invoice, which can be received by the Oracle E-Business Suite adapter.
Preconfigured?	<p>No. You must create a trading partner contract, a perimeter server, and configurations of these services or adapters:</p> <ul style="list-style-type: none">◆ Oracle E-Business Suite Adapter Configuration service◆ HTTP Server adapter◆ BP Fault Log adapter
Requires third party files?	OAG DTDs must be checked in to Application for each document type exchanged with Oracle.
Platform availability	All Application supported platforms
Related services	<ul style="list-style-type: none">◆ Oracle E-Business Suite Adapter Configuration service◆ Oracle E-Business Suite Message service◆ Timezone Offset service <p>This adapter also uses other Application services including, but not limited to:</p> <ul style="list-style-type: none">◆ Document XPath Replace service◆ Correlation service◆ Request Response XREF service◆ Request Response DXREF service◆ B2B Send adapter◆ Consume service

Application requirements	<p>You must:</p> <ul style="list-style-type: none"> ◆ Have a valid user name and password for the Oracle E-Business Suite ◆ Create a trading profile to send messages to Oracle ◆ Exchange digital certificates with Oracle E-Business suite (to use HTTPS for secure transport)
Initiates business processes?	The HTTP Server adapter must be configured with a URL that runs the OracleEBusinessReceive business process when a message is received from Oracle.
Invocation	<p>To send a message to Oracle, create a business process that does the following:</p> <ol style="list-style-type: none"> 1 Starts the Oracle E-Business Suite Adapter Configuration service. 2 Places an OAG XML document (for example, PROCESS PO) into the primary document. 3 Runs the OracleEBusinessSend business process. <p>To receive a message from Oracle, do the following:</p> <ol style="list-style-type: none"> 1 Configure a URL in the HTTP Server adapter that starts the OracleEBusinessReceive. 2 Write a single business process to handle all incoming Oracle messages, or write a separate business process for each type of incoming message (PO, Invoice, and so forth). 3 Use the Oracle E-Business Suite Adapter Configuration service to configure the option you select.
Business process context considerations	See Invocation.
Returned status values	<ul style="list-style-type: none"> ◆ Success: All steps of the business process completed successfully. If a response (Confirm BOD) indicates an application-level error, the business process status is still Success. ◆ Error: Some step in the business process had an error. The adapter BPML logs all errors to a file in a configurable folder.
Restrictions	Oracle E-Business Suite 12.0.0 and the adapter both support only UTF-8 encoding.
Persistence level	All Oracle E-Business adapter business processes use the system default persistence level (Full persistence, unless you change the system default). The Oracle E-Business Suite Adapter Configuration service, Oracle E-Business Suite Message service, and Timezone Offset service use the default service persistence level (Full persistence).
Testing considerations	Modify the sample BPML provided (samples/oracleEBusiness/bpml/testOraclePO.bpml), and import and modify the sample trading partner contract provided (samples/oracleEBusiness/TradPartnerProfiles/SendingContract.xml). A sample OAG document (samples/oracleEBusiness/OAG_XML/process_po.xml) can also be used.

Requirements

You must have a valid user name and password to log in to the Oracle E-Business Suite. To use HTTPS, you must exchange digital certificates with Oracle E-Business Suite.

Note: Modules within the Oracle E-Business Suite may use different versions of the OAG documents; be sure to use the correct version of the OAG documents for the module you use.

How the Oracle E-Business Suite Adapter Works

The adapter communicates with the Oracle E-Business Suite using the Oracle XML Gateway and Oracle Transport Agent (OTA). The OTA protocol sends and receives several parameters in addition to the OAG XML document. All of these parameters are transmitted in the form of a standard HTTP POST.

The adapter can be used to send OAG documents to Oracle and receive OAG documents from Oracle. The following topics describe how each mode works:

Sending Documents to Oracle on page 1061

Receiving Documents from Oracle on page 1062

The adapter uses a customized configuration of the BP Fault Log adapter called OracleEBusinessAdapterLogger to log all errors.

Sending Documents to Oracle

To send documents to Oracle, you must create a configuration of the Oracle E-Business Suite Adapter Configuration service that will provide the necessary parameter settings to your business process.

1. Create a business process which does the following:
 - ◆ Generates an OAG document and stores it in the primary document.
The document must have a CONFIRMATION value of 0 (no confirmation requested) or 2 (confirmation always requested); the value 1 (confirmation only on error) is not permitted.
 - ◆ Starts a configuration of the Oracle E-Business Suite Adapter Configuration service to access the configuration parameters for the adapter.
 - ◆ Runs the business process OracleEBusinessSend.
If the REFERENCEID field in the CNTROLAREA of the document is blank, the Oracle E-Business Suite adapter will set a unique value for this field; otherwise, the adapter will leave the existing value in place.

Note: This value is used to correlate responses with requests, so a unique value is required.

2. If the Oracle E-Business Suite Adapter Configuration service parameter Insert Date and Time into OAG doc at run time is set to true, the Oracle E-Business Suite adapter starts the business process OracleEBusinessInsertDate to insert the current date and time into the OAG DATETIME element within the CNTROLAREA. It will not insert the date/time into the DATETIME element within the DATAAREA of the document.
3. The Oracle E-Business Suite adapter uses the Correlation service to correlate the primary document using the names and values listed in the following table. This enables you to easily locate all Oracle OAG documents using the Correlation Search page in Application.

Correlation Name	Correlation Value	Literal or from OAG Document?
OracleEBusiness	True	Literal
ReferenceID	CNTROLAREA/SENDER/REFERENCEID	From document
Verb	CNTROLAREA/BSR/VERB	From document

Correlation Name	Correlation Value	Literal or from OAG Document?
Noun	CNTROLAREA/BSR/NOUN	From document

- The Oracle E-Business Suite adapter constructs and encodes the message using the Oracle E-Business Suite Message service.
- The Oracle E-Business Suite adapter sends the message to Oracle using the settings defined in the trading partner contract specified in the Oracle E-Business Suite Adapter Configuration service.
- If the OAG document CONFIRMATION value is 2, the Oracle E-Business Suite adapter waits for Oracle to respond with a Confirm BOD message. The Confirm BOD is placed in the primary document.

Receiving Documents from Oracle

- The Oracle E-Business Suite sends an OAG XML document to Application. The HTTP Server adapter must have a URI configured that matches the URI where the document was posted and runs the OracleEBusinessReceive business process.
- The OracleEBusinessReceive business process sends back a synchronous HTTP response to Oracle to close the HTTP connection.
- The OracleEBusinessReceive business process starts a configuration of the Oracle E-Business Suite Adapter Configuration service named OracleEBusinessAdapterConfig to get the configuration settings.

Note: If you create a configuration of this service with a different name, you must change the business process to point to the new configuration.

- The adapter decodes the message using the Oracle E-Business Suite Message service and puts the OAG XML payload into the primary document.
- The Oracle E-Business Suite adapter uses the Correlation service to correlate the primary document using the names/values listed in the previous table. This enables you to find Oracle OAG documents using the Correlation Search page in Application.

Correlation Name	Correlation Value	Literal or from OAG Document?
OracleEBusiness	True	Literal
ReferenceID	CNTROLAREA/SENDER/REFERENCEID	From document
Verb	CNTROLAREA/BSR/VERB	From document
Noun	CNTROLAREA/BSR/NOUN	From document

- If the received message is a confirm BOD, the Oracle E-Business Suite adapter sends the confirm BOD to the configuration of the OracleEBusinessSend business process that is waiting for it.

In the Confirm BOD, the value in element DATAAREA/CONFIRM_BOD/CONFIRM/CNTROLAREA/SENDER/REFERENCEID must match the value from the original request in element CNTROLAREA/SENDER/REFERENCEID

7. If the received message is not a confirm BOD, the adapter runs a business process to handle the message. The business process name is chosen as follows:
 - a. If the configuration option Business Process to handle received messages is specified, then this business process runs.
 - b. If this configuration option is left blank [Not applicable], the adapter starts a business process named OracleEBusinessRecv_<VERB>_<NOUN>, where <VERB> and <NOUN> are taken from the received OAG document CNTROLAREA. For example, a PROCESS PO would be handled by a business process named OracleEBusinessRecv_PROCESS_PO.
8. If the configuration service parameter Automatically send Confirm BOD for received messages is false, confirm BOD is returned to Oracle if required. If this parameter is true, the adapter will send back a confirm BOD using the business process OracleEBusinessSendConfirmBOD if either of the following conditions is met:
 - a. The CONFIRMATION value in the received document was 2.
 - b. The CONFIRMATION value in the received document was 1 and there was an error handling the received document, either in an adapter business process or a user-defined sub-process called to handle the received message.

Message Exchange

The Oracle E-Business Suite adapter supports the OAG standard XML documents. The OAG provides DTDs (and XML schema, in OAG version 8.0) to define the documents which must be checked into Application. The following DTDs are already checked in; you must check in any others (including different versions of these transactions).

- 002_confirm_bod_004.dtd
- 003_process_po_007.dtd
- 161_show_shipment_005.dtd
- 171_process_invoice_002.dtd
- oagis_domains.dtd
- oagis_entity_extensions.dtd
- oagis_extensions.dtd
- oagis_fields.dtd
- oagis_resources.dtd
- oagis_segments.dtd

The adapter and Oracle XML Gateway support only UTF-8 encoding. For any other encoding, use the Encoding service to change encoding to UTF-8. See *Encoding Conversion Service* for information about this service.

Implementing the Oracle E-Business Suite Adapter

To implement the Oracle E-Business Suite adapter, complete the following tasks:

1. Activate your license for the Oracle E-Business Suite adapter. For information, see *An Overview of Implementing Services*.
2. Check in the DTDs or XML schema for the OAG documents you will exchange. Several OAG DTDs are pre-installed with the adapter. For information, see *Message Exchange* on page 1063.
3. Configure the configuration of the BP Fault Log adapter named OracleEBusinessAdapterLogger. Set the path to the log file (for example, /home/sampleuser/logs/oracleEBusiness.log). See *BP Fault Log Adapter* for more information about the BP Fault Log adapter.
4. If using HTTPS for secure transport, exchange digital certificates between Application and Oracle E-Business Suite, and configure the web servers on both sides to use the certificates. The steps required will depend on the web server used on both sides and whether you use self-signed or CA-signed certificates.

For the Oracle configuration, see the following documents (152775.1 and 119873.1). You will need an Oracle Metalink account to view these documents.

- ◆ <http://metalink.oracle.com/metalink/plsql/showdoc?db=Not&id=152775.1>
- ◆ http://metalink.oracle.com/metalink/plsql/ml2_documents.showNOT?p_id=119873.1

5. Create and configure a perimeter server for use with the Oracle E-Business Suite adapter.
6. Create an HTTP Server adapter configuration called Oracle_HTTPServerAdapter. For information about configuring the HTTP Server adapter, see *HTTP Server Adapter*. To set up this configuration, complete the following steps:
 - a. From the **Administration** menu, select **Deployment > Services > Configuration**.
 - b. Select **Create > New Service**.
 - c. For Service Type, choose **HTTP Server Adapter**, then click **Next**.
 - d. Complete the adapter configuration using the settings shown in the table below for the corresponding fields. Configure all other fields according to your requirements:

For this field...	Specify this setting...
Name	Oracle_HTTPServerAdapter
Perimeter Server Name	Select the node and perimeter server you configured in step 5.
URI	1. On the Oracle_HTTPServerAdapter: URI page, select add New URI . 2. The Oracle_HTTPServerAdapter: URI: URI Config page displays. Enter the URI where Oracle will post messages. This is the interface through which Oracle E-Business Suite will call Application. Example: /b2bhttp/inbound/oracleadapter
Launch BP or WAR	Select Business Process .
Business Process	Choose OracleEBusinessReceive
Send Raw Messages	Select No .

- e. Deploy the perimeter server on the host machine in the DMZ if required.

7. Create the following trading profiles for Application and Oracle:
 - a. Application consumption profile for receiving messages from Oracle E-Business Suite. This profile is not used; it is required only because a contract requires both a production and consumption profile.
 - b. Oracle E-Business Suite production profile for sending messages to Oracle E-Business Suite
 You can configure the trading profiles by importing the SendingContract.xml file in the samples/oracleEBusiness/TradPartnerProfiles directory and modifying the trading partner profiles. For more information, see *Editing the Trading Profile Sample* on page 1068.
8. Create one or more business processes to handle messages received from Oracle. You may create a single process to handle all incoming messages, or a separate business process for each type of message received, based on the NOUN/VERB combination in the received OAG document. This is done as part of the Oracle E-Business Suite Adapter Configuration service setup (option: Business Process to handle received messages). See *Receiving Documents from Oracle* on page 1062 for more information.

If you choose to use a different business process for each message type, the process names must be OracleEBusinessRecv_<VERB>_<NOUN>.

9. If you select the option Automatically send Confirm BOD for received messages, you must create a template Confirm BOD. A sample Confirm BOD is provided in the directory samples/oracleEBusiness/OAG_XML/ConfirmBODtemplate.xml. For more information, see *Creating a Confirm BOD template* on page 1069.
10. In Application, set up the configuration of the Oracle E-Business Suite Adapter Configuration service named OracleEBusinessAdapterConfig. Choose the trading partner contract created earlier in this process.
11. Create a business process that starts the OracleEBusinessSend business process. Your business process must include a call to your configuration of the Oracle E-Business Suite Adapter Configuration service, and it must put the OAG XML document into the primary document. See testOraclePO.bpml in the directory samples/oracleEBusiness/bpml for an example.

Oracle uses the following message parameters to match a message to a transaction defined in the XML Gateway. The adapter assigns these parameters based on values in the OAG document as shown in the table below. The assignments occur when the OracleEBusinessSend business process starts the Oracle E-Business Suite Message service.

Message Parameter Name	Must match XML Gateway transaction field	Adapter assigns parameter value from OAG element
TRANSACTION_TYPE	External transaction type	CNTROLAREA/BSR/NOUN
TRANSACTION_SUBTYPE	External transaction subtype	CNTROLAREA/BSR/NOUN
PARTY_SITE_ID	Source location code	CNTROLAREA/SENDER/LOGICALID

To assign different values to any of these fields, the user-defined business process must assign values to the fields. For example: <assign to="TRANSACTION_TYPE">MyTransactionType</assign>

12. Run the business process.

Oracle E-Business Suite Custom Services

The Oracle E-Business Suite adapter uses the following custom services, which are installed automatically with the adapter. This section contains a brief overview of how each service is used, but specific information about each can be found on the pages listed below:

Oracle E-Business Suite Adapter Configuration service: for more information, see *Oracle E-Business Suite Adapter Configuration Service* on page 1066.

Oracle E-Business Suite Message service: for more information, see *Oracle E-Business Suite Message Service* on page 1066.

Timezone Offset service: for more information, see *Timezone Offset Service* on page 1067.

For general information about creating service configurations, see *Managing Services and Adapters*.

Oracle E-Business Suite Adapter Configuration Service

Use the Oracle E-Business Suite Adapter Configuration service to set configuration parameters for the adapter.

You must create a configuration of the Oracle E-Business Suite Adapter Configuration service and start it from the business process which runs the OracleEBusinessSend business process to send messages. For receiving messages, the OracleEBusinessReceive business process starts an configuration of this service named OracleEBusinessAdapterConfig. In a typical situation, it is best to create a single configuration of this service and name it OracleEBusinessAdapterConfig.

At run time, this service simply inserts the configured parameters into process data. The configured parameter values cannot be overridden with the GPM.

Oracle E-Business Suite Message Service

This service encodes messages for sending to Oracle and decodes messages received from Oracle. The Oracle XML Gateway/Oracle Transport Agent (OTA) uses several parameters in addition to the XML payload, and it uses standard HTTP POST semantics for all parameters including the XML payload.

See the business process OracleEBusinessSend for an example of using this service to encode a message for sending. To encode a message, the XML payload must be in the primary document when this service runs. The resulting encoded message (in standard HTTP POST format) will be placed in the primary document as output.

See the business process OracleEBusinessReceive for an example of using this service to decode a received message. The XML payload will be placed in the primary document as output.

Service Output for ENCODE MODE

With the MODE set to ENCODE, this service outputs to the primary document only. The primary document will contain an HTTP POST message ready to be sent to Oracle.

Service Output for DECODE MODE

When this service runs with the MODE set to DECODE, the following output will be produced:

The PAYLOAD (XML document) of the received message will be stored in the primary document.

All other OTA parameters except the PASSWORD will be stored in process data within the element ProcessData/OracleEBusiness/Recv_OTA_Params, for example ProcessData/OracleEBusiness/Recv_OTA_Params/USERNAME.

The OTA parameter PASSWORD will not be stored for security reasons.

Timezone Offset Service

This service computes the difference between the local time zone and Greenwich Mean Time (GMT), using the format +/-HHMM. For example, the value for U.S. Eastern time would be -0500 during standard time, or -0400 during daylight savings time.

Output

The output of this service is a single XML element named timezoneOffsetFromGMT.

Predefined Business Processes

The Oracle E-Business Suite adapter uses predefined business processes to handle communications with Oracle.

The following table describes each of the predefined business processes:

Business Process	Description
OracleEBusinessSend	The top-level business process to send a message to Oracle. <ol style="list-style-type: none">1 If REFERENCEID is blank, assign it a unique value.2 If the Insert date and time option is true, run the business process OracleEBusinessInsertDate3 Run the business process OracleEBusinessCorrelate4 Encode the XML document using the Oracle E-Business Suite Message service5 Send the message to Oracle E-Business Suite6 If a Confirm BOD is expected, wait for it
OracleEBusinessInsertDate	Replace the values within the CNTRLAREA/DATETIME element of the primary document with the current time.
OracleEBusinessCorrelate	Uses the Correlation service to correlate the primary document. See the correlation table in the section <i>Sending Documents to Oracle</i> on page 1061 for a list of the values correlated.

Business Process	Description
OracleEBusinessReceive	<p>The top-level business process to handle all messages received from E-Business Suite.</p> <ol style="list-style-type: none"> 1 Run the business process OracleEBusinessCorrelate 2 If the received message is a Confirm BOD, send the Confirm BOD to the configuration of OracleEBusinessSend which is waiting for it. 3 If the message is not a Confirm BOD, run the user-specified business process to handle it. If you did not specify a business process, run a business process named OracleEBusinessRecv_<VERB>_<NOUN> 4 If the Auto send Confirm BOD option is set to true, run the business process OracleEBusinessSendConfirmBOD
OracleEBusinessSendConfirmBOD	<p>Sends a Confirm BOD message to Oracle. If the Insert date and time option is true, run the business process OracleEBusinessInsertDate to insert the current date into the Confirm BOD CNTROLAREA/DATETIME</p>

Editing the Trading Profile Sample

You can configure the trading profiles by importing the profile XML file in the samples/oracleEBusiness/TradPartnerProfiles directory and then editing parts of the profile.

To edit the sample trading profile:

1. Import the sample trading profile, SendingContract.xml, into Application. At the prompt for the passphrase, enter **password**.
2. If you are using HTTPS, obtain the digital certificate used by the Oracle E-Business Suite. Check this certificate in to Application as both a trusted certificate and a CA certificate.
3. Edit the Oracle EBusiness Suite transport:

In the **End Point** field, enter the URL for the Oracle XML gateway.

Note: Note that the protocol part of the URL will be HTTP whether you use HTTP or HTTPS.

If you are using HTTPS:

- ◆ For **SSL**, select **MUST**.
- ◆ For **Cipher Strength**, select **STRONG**.
- ◆ In the **User Certificate** field, select the trusted certificate from Oracle E-Business Suite.
- ◆ In the **CA certificate** field, select the CA certificate from Oracle E-Business Suite.
- ◆ In the **Key Certificate** field, select the Application certificate (B2BHttp).

If you are using HTTP:

- ◆ For **SSL**, select **None**.
- ◆ All three certificate fields can be left blank.

Creating a Confirm BOD template

If you select Automatically send Confirm BOD for Received Messages, then you must create a template Confirm BOD file. A sample Confirm BOD is provided in the directory `samples/oracleEBusiness/OAG_XML/ConfirmBODtemplate.xml`.

Consider the following when creating a Confirm BOD template:

If the first REFERENCEID element (`/CONFIRM_BOD_004/CNTROLAREA/SENDER/REFERENCEID`) is left blank, the OracleEBusinessSendConfirmBOD business process will insert a unique value for this element. Typically, Oracle will expect a unique value here, so it is recommended to leave this element blank.

The first CONFIRMATION value (`/CONFIRM_BOD_004/CNTROLAREA/SENDER/CONFIRMATION`) should be set to 0 (zero) or 1, but not 2. The value 2 indicates that a Confirm BOD is requested, but it does not make sense to request a Confirm BOD in response to a Confirm BOD.

The OracleEBusinessSendConfirmBOD business process will replace all the confirm BOD elements within `DATAAREA/CONFIRM_BOD/CONFIRM` with the `CNTROLAREA` from the original request document.

BPML Error Handling

The adapter logs errors in a file specified by the OracleEBusinessAdapterLogger configuration. Errors are also passed to the user-defined business process that ran the OracleEBusinessSend business process. During error handling, the adapter cleans out any waiting consume services and unusable XREF information. This ensures that when you rerun the adapter nothing interferes with its operation.

Security

The adapter supports HTTPS/SSL for secure transport. The adapter also hides the Oracle E-Business Suite password so the clear text password is not shown in process data.

Internationalization

The adapter supports internationalization using UTF-8 encoding. This is the only encoding supported by Oracle E-Business Suite and XML Gateway. If your documents use any other encoding, use the Encoding Conversion service to convert to UTF-8. See *Encoding Conversion Service* for information about this service.

Adapter for PeopleSoft

The Adapter for PeopleSoft® interacts with the PeopleSoft system, either by requesting data or specifying an operation to be performed. The Adapter for PeopleSoft uses three business processes to interact with the PeopleSoft system:

PeopleSoft Send (PS Send) – Application sends data to PeopleSoft. No response from PeopleSoft is necessary.

PeopleSoft Send and Wait (PS Send and Wait) – Application sends a request to PeopleSoft, receives a response from PeopleSoft, and ties the response back to the waiting business process.

PeopleSoft Receive (PS Receive) – Application receives asynchronous response messages from PeopleSoft and starts the PS Receive business process. The PS Receive process forwards the response to the PS Send and Wait business process that generated the request.

Note: This adapter is certified for use with PeopleSoft 8.1 and 8.4.

Note: This adapter works with Perimeter Services.

The following table provides an overview of the Adapter for PeopleSoft:

System name	None
Graphical Process Modeler (GPM) category	None
Description	<p>Set of business processes used within Application to facilitate communication with a PeopleSoft system. The Adapter for PeopleSoft uses the HTTP Server adapter, HTTP Client Begin Session service, HTTP Client POST service, HTTP Client End Session service, HTTP Client adapter (used by the HTTP Client services), HTTP Respond service, and some of the Application internal services (for example, RequestResponseXREF and RequestResponseDXREF) to post messages to a PeopleSoft system and to receive asynchronous responses. The responses are matched against pending requests using XREF and DXREF services.</p> <p>The ZlibInflate service and TimeStamp Service for PeopleSoft are part of the Adapter for PeopleSoft. The ZlibInflate service is used to uncompress messages received from PeopleSoft, if necessary. The Timestamp Service for PeopleSoft is used to insert a timestamp in a PeopleSoft-specific format within acknowledgements sent to PeopleSoft for responses received.</p>

Business usage	<p>The Adapter for PeopleSoft is a set of business processes, each representing a single operation:</p> <ul style="list-style-type: none"> ◆ The PS_Send business process posts a message to PeopleSoft without expecting any response beyond an immediate acknowledgement. ◆ The PS_Send_Wait business process posts a message to PeopleSoft and waits inside a Consume service till it receives a response to its request. ◆ The PS Receive business process is initiated by the HTTP Server adapter when a message is received from PeopleSoft. If the message received is a response to a pending request, the PS Receive business process produces a message for consumption by the appropriate PS Send Wait process that created that request. You must create configurations of the HTTP Server adapter, HTTP Respond service, ZlibInflate service, and PeopleSoft Time Stamp for use by the PS Receive business process.
Usage example	<p>The PS Send business process can be used to post messages such as a shipping receipt notice to PeopleSoft. No response is expected beyond an immediate acknowledgement that this message was received.</p> <p>The PS Send Wait business process can be used to post requests such as a sales order load. In this case, a response is expected from PeopleSoft, and the business process does not continue till the response is received.</p> <p>The HTTP Server adapter can be configured to initiate the PS Receive business process when an asynchronous message is received from PeopleSoft. Depending on the type of message received, the PS Receive business process may produce a message for a waiting PS Send Wait business process.</p>
Preconfigured?	No
Requires third party files?	<p>The Adapter for PeopleSoft adapter requires the installation of PeopleSoft version 8.1, 8.4, or 8.8 using the 8.1 connector.</p> <p>Note: DTDs and schemas are available from PeopleSoft to meet mapping needs.</p>
Platform availability	All supported Application platforms
Related services	<ul style="list-style-type: none"> ◆ HTTP Server adapter ◆ HTTP Respond service ◆ HTTP Client Begin Session service ◆ HTTP Client POST service ◆ HTTP Client End Session service ◆ RequestResponseXREF service ◆ RequestResponse DXREF service ◆ Produce service ◆ Consume service <p>The adapter also requires a Perimeter server.</p>
Application requirements	<p>You must create an input XML document that represents (or may be pre-processed into) valid PeopleSoft messages or requests. The document will serve as input to the PS Send and PS Send And Wait business processes.</p>

Initiates business processes?	You can initiate a PS Receive business process through an HTTP Server adapter to receive asynchronous response messages from PeopleSoft.
Invocation	Not applicable
Business process context considerations	None
Returned status values	Basic Status: Success or Failure Advanced Statuses: None
Restrictions	You can use the Adapter for PeopleSoft business processes to perform these operations only: <ul style="list-style-type: none"> ◆ Send ◆ Send and Wait ◆ Receive an asynchronous message to a Application request Autonomous asynchronous messages (other than ping) from PeopleSoft are not supported.

Requirements

To configure and use the Adapter for PeopleSoft, you should know how to:

- Use the Application Map Editor.

- Create and edit a business process using the Application GPM.

For the Adapter for PeopleSoft to work correctly, you must establish an HTTP connection to PeopleSoft from Application at run time.

How the Adapter for PeopleSoft Works

To understand how the Adapter for PeopleSoft works, you need to understand how the business processes associated with the adapter interact with the PeopleSoft system.

PeopleSoft Send Business Process (PS Send)

You can run the PeopleSoft Send (PS Send) business process manually or from a parent business process. The PS Send business process sends an HTTP post to the PeopleSoft system.

The following components must be in place to connect a business process through HTTP:

- HTTP Begin Session service, which initiates the connection to PeopleSoft through the Perimeter Server

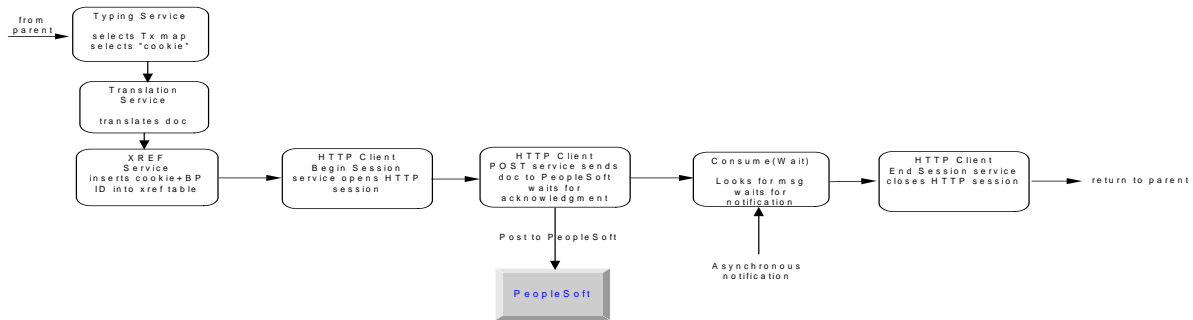
- Perimeter Server, which listens for connection requests

- HTTP Client adapter, which communicates with PeopleSoft

- HTTP Client POST service, which sends the HTTP POST request

- HTTP Client End Session service, which closes the session with PeopleSoft

The following figure shows how the PS Send business process works:



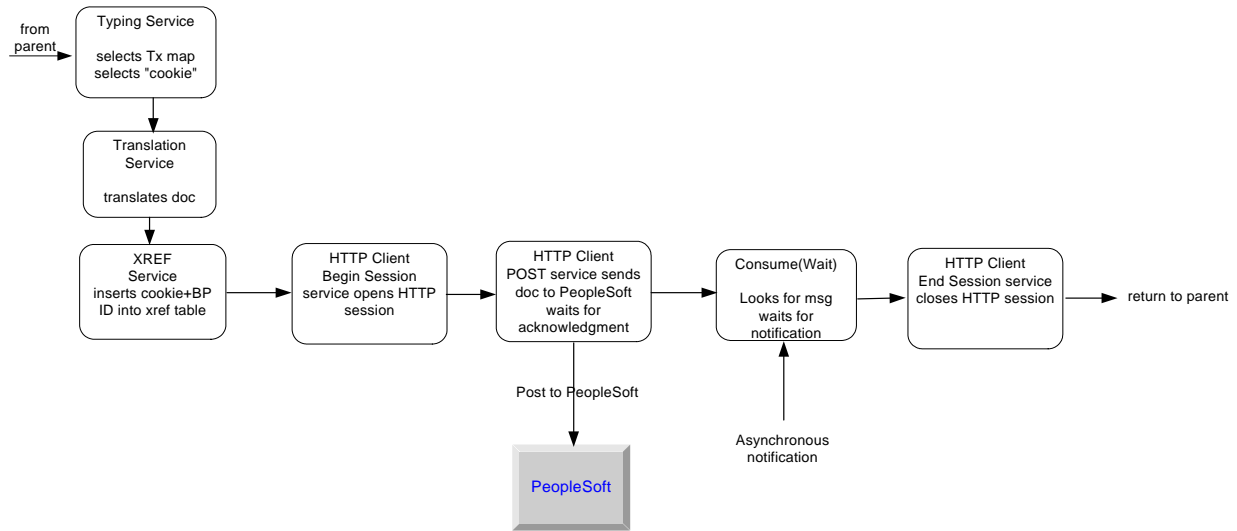
As it runs, the PS Send business process performs the following steps:

1. Determines the type of message being sent
2. Translates the input message to a PeopleSoft-supported message
3. Sends an HTTP post to the PeopleSoft system, using the HTTP Client services and adapter.

PeopleSoft Send and Wait Business Process (PS Send and Wait)

The PeopleSoft (PS) Send and Wait business process works the same as the PS Send business process with an additional function: PS Send and Wait waits for an asynchronous notification from another business process.

The following figure shows how the PS Send and Wait business process works:



For information about how the PS Send and Wait business process uses Application services to perform its work, see *Implementing the PS Send and Wait Business Process* on page 1081.

PeopleSoft Receive Business Process (PS Receive)

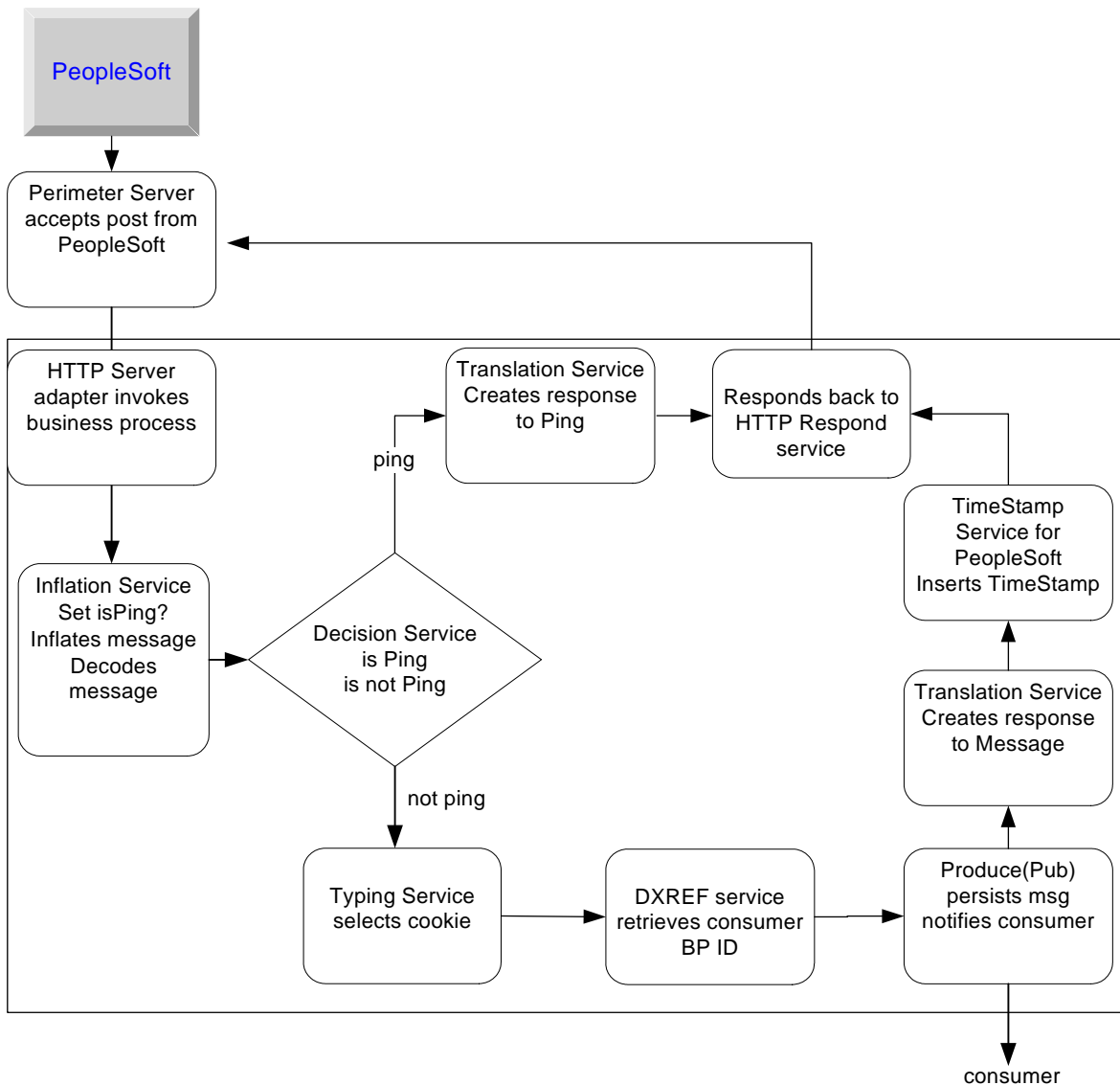
The PeopleSoft Receive (PS Receive) business process uses the HTTP Server adapter and HTTP Respond service to perform its functions. PeopleSoft never communicates directly with the PeopleSoft Receive business process. The HTTP Server adapter receives the communication and passes the information to the business process. Responses to PeopleSoft, if required, are done using the HTTP Respond service.

Note: The HTTP Respond service and HTTP Server adapter are used together to receive messages from a trading partner and to send back a response on the same connection. See *HTTP Server Adapter* and *HTTP Respond Service*.

The following components must be in place to connect a business process through HTTP:

- Perimeter server, which listens for a message
- HTTP Server adapter, which initiates the business process
- HTTP Respond service, which sends back a response to PeopleSoft

The following figure shows how the PS Receive process works:



Upon receiving an HTTP post, the Perimeter server notifies the HTTP Server adapter, which starts the PS Receive business process. After it has started, PS Receive notifies a business process waiting for a response from PeopleSoft.

To complete the notification, PS Receive:

1. Decodes and inflates the incoming post
2. Determines which type of message was received

3. Makes a decision based upon the message type:

If the Message Is a	Then PS Receive
Simple ping request from PeopleSoft	Sends acknowledgment back to the HTTP Respond service so that it can notify PeopleSoft.
PeopleSoft response to a previous request	Publishes a notification to the waiting business process and responds to the HTTP Respond service to acknowledge the correct processing of the PeopleSoft incoming message.

4. Creates a response and sends it to the HTTP Respond service so that it can update PeopleSoft.

For information about how the PS Receive business process uses Application services to perform its work, see *Implementing the PS Receive Business Process* on page 1084.

Implementing the Adapter for PeopleSoft

To implement the Adapter for PeopleSoft, complete the following tasks:

1. Activate your license for the Adapter for PeopleSoft. See *An Overview of Implementing Services*.
2. Implement the following business processes, as needed:
 - ◆ PS Send business process. See *Implementing the PS Send Business Process* on page 1076.
 - ◆ PS Send and Wait business process. See *Implementing the PS Send and Wait Business Process* on page 1081.
 - ◆ PS Receive business process. See *Implementing the PS Receive Business Process* on page 1084.

Implementing the PS Send Business Process

To implement the PS Send business process, you must set up the business process properly to work with the following Application services:

Typing service
Translation service
HTTP Client Begin Session service
HTTP Client POST service
HTTP End Session service

Typing Service for PS Send

The Typing service enables Application to determine, at run time, which translation to perform on the input document. The service determines the input document type and sets a business process parameter (map_name) indicating which map to use to translate the document.

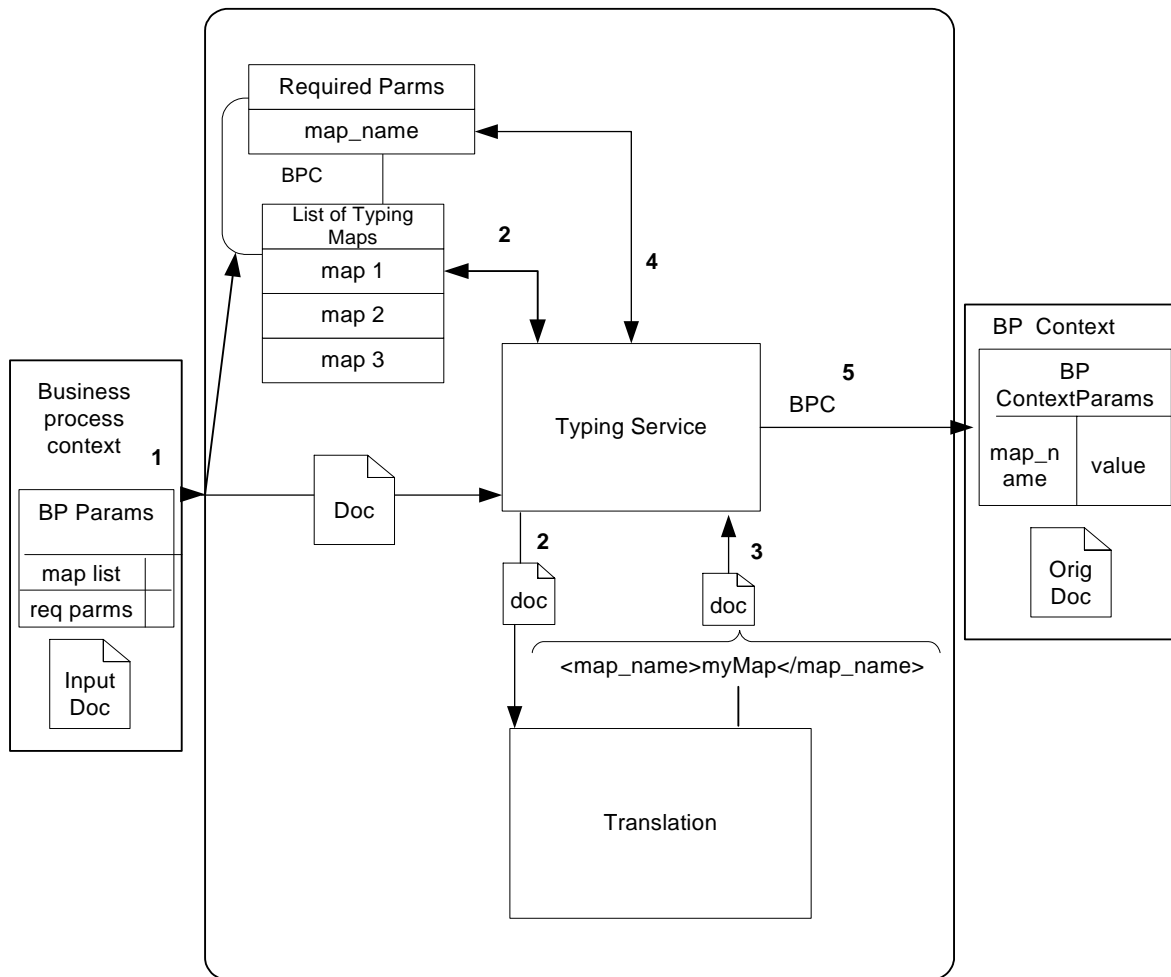
For the Typing service to work with PS Send, you need to set two parameters in the Typing service BPML:

typing_maplist – This is a space-separated list of typing maps that the Typing service uses to determine the document type. In this list, include a typing map for each PeopleSoft-supported transaction.

required_parmlist – This is a space-separated list of parameters that the Typing service sets in the business process. map_name is a required parameter in this list.

How the Typing Service Works

The following figure shows how the Typing service works:



1. When the Typing service receives an input document, the Typing service takes the first map in the typing_maplist parameter and attempts to translate the input document using that map.
2. If the input document matches the left side of the map, the Typing service uses the Translation service classes to perform the translation defined by the map.

If the input document does not match the left side of the map, the Typing service takes the next map in the typing_maplist parameter and attempts a translation. This process continues until a match is found and a translation occurs.

3. At translation time the following extended rule executes:

```
#map_name="fixed_sales_order_load"
```

This rule sets the `map_name` element value to *fixed_sales_order_load* in the output of the translation.

4. The Typing service searches the output of the document translation for the `map_name` element, which is specified in the `required_parmlist` parameter.
5. The Typing service takes the value of the specified element and sets a business process parameter using the name from the `required_parmlist` and the value found in the element. For example, `param name = map_name, value = fixed_sales_order_load`.

Creating a Typing Map for PS Send

To determine the input document type, the Typing service uses a Application typing map.

Before using the PS Send business process, you must create a typing map in the Application Map Editor for each message supported by the business process. The following procedure assumes that you have a working knowledge of the Application Map Editor.

1. Verify that:
 - a. The DTDs on the right side of the map have element tags corresponding to the business process parameters to be set.
 - b. The left side of the map corresponds to the DTD for the input message to be supported.
 - c. The right side of the map contains `<map_name></map_name>`.
2. Set an On Begin extended rule for the `map_name` element on the right side. The rule should look like this:

```
#map_name="fixed_sales_order_load";
```
3. Link a required element on the left side to the `map_name` on the right side. The left side input element, mapped to the right side output element (`map_name`), must contain a value. If no value exists, the transaction process will not proceed correctly.

Configuring the Typing Service for PS Send

To configure the Typing service for PS Send:

1. Create a typing map (left side = input message DTD, right side = `map_name` DTD).
2. Check the map in to Application.
3. Create a Typing Service configuration. See *Creating a Service Configuration*.
4. Set the parameters `typing_maplist` and `required_parmlist` in the service configuration.

The following code is an example of the Typing service BPML for the PS Send business process:

```
<operation name="Typing">
  <participant name="TypingService"/>
  <output message="Xout">
    <assign to="." from="*" />
    <assign to="typing_maplist">SalesOrderLoadMapName SalesQuoteLoadMapName
ShippingNotificationMapName</assign>
    <assign to="required_parmlist">map_name</assign>
    <assign to="validate_input_against_dtd">yes</assign>
  </output>
```

```
<input message="Xin">
<assign to="." from="*" />
</input>
</operation>
```

Translation Service for PS Send

The Translation service receives the input document from the Typing service. Using the map indicated by the `map_name` business process parameter, the Translation service translates an Application-defined document to a PeopleSoft-supported message.

Typically, to use the Translation service, you would include the following rule in the BPML:

```
<assign to="map_name">myMapName</assign>
```

Because the Typing service sets the business process parameter for you, an assign rule is not needed.

Creating a Translation Map for PS Send

To translate the input document, the Translation service uses an Application map. Before using the business process, you must create a translation map in the Application Map Editor for each message supported by the business process. This map is created in much the same way as a typing map. For more information, see *Creating a Typing Map for PS Send* on page 1078.

Configuring the Translation Service for PS Send

To configure the Translation service for PS Send:

1. Create a translation map (left side = input message DTD, right side = PS supported msg DTD).
2. Check the map in to Application.
3. Create a Translation service configuration. See *Creating a Service Configuration*. No BPML parameters are required.

The following code is an example of the Translation service BPML for the PS Send business process.

```
<operation name="translate">
  <participant name="Translation"/>
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

HTTP Client Services and Adapter for PS Send

The HTTP Client Begin Session service initiates an HTTP session with PeopleSoft through the HTTP Client adapter. After the session is initiated, the HTTP Client POST service takes the translated document and posts it to a PeopleSoft system through HTTP. The HTTP Client End Session service closes the HTTP session.

Configuring the HTTP Client Services for PS Send

To configure the HTTP Client services for the PS Send business process:

Create configurations of the HTTP Client Begin Session service, HTTP Client POST service, and HTTP End Session service. See *Creating a Service Configuration*.

The following code is an example of the HTTP Client services BPML for PS Send:

```
<sequence name="send data">
  <operation name="HTTP Client Begin">
    <participant name="HTTPClientBeginSession"/>
    <output message="HttpClientBeginServiceInputMessage">
      <assign to="." from="PrimaryDocument"/>
      <assign to="RemoteURL">http://remote/url/to/peoplesoft</assign>
      <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
  <operation name="HTTP Client Post">
    <participant name="HTTPClientPost"/>
    <output message="HttpClientPostServiceInputMessage">
      <assign to="." from="PrimaryDocument"/>
      <assign to="SessionToken" from="SessionToken/text()" />
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
  <operation name="HTTP Client End">
    <participant name="HTTPClientEndSession"/>
    <output message="HttpClientEndServiceInputMessage">
      <assign to="SessionToken" from="SessionToken/text()" />
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
  <onFault>
    <sequence>
      <operation name="HTTP Client End">
        <participant name="HTTPClientEndSession"/>
        <output message="HttpClientEndServiceInputMessage">
          <assign to="SessionToken" from="SessionToken/text()" />
        </output>
        <input message="inmsg">
          <assign to="." from="*" />
        </input>
      </operation>
    </sequence>
  </onFault>
</sequence>
```

When it has been configured properly, the PS Send business process can support almost any PeopleSoft-supported XML message.

Implementing the PS Send and Wait Business Process

To implement the PS Send and Wait business process, you must set up the business process properly to work with the following Application services:

- Typing service
- Translation service
- Request Response XREF service
- HTTP Client Begin Session service
- HTTP Client POST service
- HTTP Client End Session service
- Consume service

Typing Service for PS Send and Wait

In the PS Send and Wait business process, the Typing service selects the correct translation map and the cross-reference (XREF) cookie information to correlate incoming PeopleSoft responses with Application requests. For example, the XREF cookie could be an invoice number or a contract number.

The Typing service extracts the XREF cookie information from the business process input document. The fields used for the XREF cookie are determined during the manual creation of the Typing map.

Follow these XREF cookie requirements:

- The business process parameter `required_parmlist` must contain the XREF cookie element.

- The parameter `required_parmlist` must exist as an element on the output (right side) of the typing map.

- The following rules enables the XREF cookie element to provide a value during translation of the typing map:

Element	Map Side (Input/Output)	Rule Type	Description	Example
map_name	Output(right)	Extended	Indicates a hard-coded map name	<code>#map_name = "fixed_sales_order_load";</code>
NA	NA	Pre-session	Declares a global variable available to all rules	<code>string[100] theBusinessUnit;</code>
theBusinessUnit	Input (left)	Extended	Gives the global variable a value	<code>theBusinessUnit= #BUSINESS_UNIT_2;</code>
XREFCookie	Output(right)	Extended	Sets the XREF cookie element to the global variable	<code>#XREFCookie =theBusinessUnit;</code>

Configuring the Typing Service for PS Send and Wait

To configure the Typing service for PS Send and Wait:

1. Create a typing map (left side = input message DTD, right side = map_name and XREF cookie DTD).

2. Create a pre-session standard rule declaring a global variable. This variable holds the XREF cookie value.
3. Create an extended rule on an element on the left side of the map. Use this element to assign a value for the global variable declared in step 2.
4. Create an extended rule for the XREF cookie element on the right side of the map. Use the global variable to assign a value to the XREF cookie element.
5. Link the required element on the left to the XREF cookie element on the right.
6. Create an extended rule on the right side map_name parameter. Assign a hard-coded value to the map_name element.
7. Link a required element on the left to the map_name element on the right.
8. Save and compile the maps.
9. Check the map in to Application.
10. Create a Typing service configuration. See *Creating a Service Configuration*.
11. Set the parameters typing_maplist and required_parmlist in the service configuration.

The following code is an example of Typing service BPML for the PS Send and Wait business process:

```
<operation name="Typing">
  <participant name="TypingService"/>
  <output message="Xout">
    <assign to="." from="*" />
    <assign to="typing_maplist">SalesOrderLoad_Map_and_XREFCookie
SalesQuoteLoad_Map_and_XREFCookie ShippingNotification_Map_and_XREFCookie</assign>
    <assign to="required_parmlist">map_name XREFCookie</assign>
    <assign to="validate_input_against_dtd">yes</assign>
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

Translation Service for PS Send and Wait

For the PS Send and Wait business process, the Translation service operates in the same manner as it does for the PS Send business process. For more information, see *Translation Service for PS Send* on page 1079.

Request Response XREF Service for PS Send and Wait

The Request Response XREF service, along with the Request Response DXREF service, provides the link between the PS Send and Wait and the PS Receive business processes.

1. The Request Response XREF service persists the PS Send and Wait business process ID, and the XREF cookie selected by the Typing service.
2. The Request Response DXREF service uses the XREF cookie to look up the business process ID of the corresponding PS Send and Wait business process.
3. When the PS Receive business process pings the PS Send and Wait business process, the PS Receive business process uses the business process ID.

Configuring the Request Response XREF Service for PS Send and Wait

To configure the Request Response XREF service for PS Send and Wait, create the service configuration.

The following code is an example of the Request Response XREF service BPML for the PS Send and Wait process.

```
<operation name="Cross Reference">
  <participant name="RequestResponseXREFService"/>
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

HTTP Client Services and Adapter for PS Send and Wait

The HTTP Client services and adapter operate in the same manner as they do for the PS Send business process. For more information, see *HTTP Client Services and Adapter for PS Send* on page 1079.

Consume Service for PS Send and Wait

The Consume service facilitates communication between the PS Send and Wait and the PS Receive business processes. The link created by the XREF and DXREF services enables this communication. To provide the consuming end of the communication, the Consume service performs the following steps:

1. The Consume service sends the business process ID to the database. This signifies to the producing business processes that the business process associated with the ID is a candidate for consuming messages.
2. The Consume service monitors existing messages and searches the database for any messages that correspond to its business process ID.
3. If there are no existing messages, the Consume service waits for notification to continue from an external business process.
4. When notified, the Consume service searches for a message in the database, and if found, submits the message into the process data.

Configuring the Consume Service for PS Send and Wait

To configure the Consume service for PS Send and Wait:

1. Create a Consume service configuration. See *Creating a Service Configuration*.
2. Set the input message attribute to the name of the Produce service output message. This name is set in the PS Receive BPML. For more information about the Produce service, see *Configuring the Produce Service for PS Receive* on page 1088.
3. Set a parameter corresponding to the Produce service parameter that holds the PeopleSoft message. This consume parameter enables the message passed to the Consume service to be placed in the process data. Additional business processes or services can also access the message.

The following code is an example of the Consume service BPML for the PS Send and Wait process:

```

<consume>
  <input message="anOutMsg">
    <assign to="." from="//aDocument"></assign>
  </input>
</consume>

```

Implementing the PS Receive Business Process

To implement the PS Receive business process, you must set up the business process properly to work with a Perimeter server and the following Application services:

- Inflation service
- Typing service
- Decision Engine service
- Request Response DXREF service
- Translation service
- Produce service
- HTTP Server adapter
- HTTP Respond service
- Timestamp Service for PeopleSoft

Inflation Service for PS Receive

The incoming message sent by PeopleSoft is compressed and encoded. To process the data on the message, Application must decode and then uncompress (inflate) the data. The Inflation service performs this action. In addition, the Inflation service contains simple typing logic to determine whether the incoming message is a ping message or a business request. The message type is used by the Decision service to handle the message appropriately.

Configuring the Inflation Service for PS Receive

No configuration is required.

The following code is an example of the Inflation service BPML for the PS Receive process:

```

<operation name="Inflate incoming PS message">
  <participant name="ExampleZlibInflateService"/>
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>

```

Typing Service for PS Receive

The PS Receive business process uses the Typing service to set the XREF cookie.

Setting the XREF Cookie

The Typing service sets the business process parameters for the XREF cookie. For more information, see *Typing Service for PS Send and Wait* on page 1081.

Configuring the Typing Service for PS Receive

To configure the Typing service for PS Receive:

1. Create a typing map (left side = input message DTD, right side = map_name and XREF cookie DTD).
2. Create a pre-session standard rule declaring a global variable. This variable holds the XREF cookie value.
3. Create an extended rule on the left side of the map to assign a value to the global variable declared in step 2.
4. Create an extended rule for the XREF cookie element on the right side of the map. Use the global variable to assign a value to the XREF cookie element.
5. Link the required element on the left to the XREF cookie element on the right.

The following code is an example of BPML for setting the XREF cookie:

```
<operation name="Typing">
  <participant name="TypingService"/>
  <output message="Xout">
    <assign to="." from="*" />
    <assign to="typing_maplist">SalesOrderAck_XREFCookie</assign>
    <assign to="required_parmlist">XREFCookie</assign>
    <assign to="validate_input_against_dtd">yes</assign>
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

If an unsolicited transaction is received by the PS_Receive adapter, the Typing service fails. The BPML unencodes and decompresses the XML document and stores it as the primary document.

To handle unsolicited transactions, add an on-fault in the Respond To Message sequence. The sequence in the on-fault can further process the transaction received. For example, you could print the document out to the file system so that another business process can collect the file and start a different process.

Decision Engine Service for PS Receive

The PS Receive business process receives two types of messages:

Simple ping

Business response to an Application-initiated request

The Decision Engine service enables the PS Receive business process to react differently to each message type. The service takes the output of the Inflation service and performs actions based on the document type. If the message is a ping, the Decision Engine service directs the process flow to create a response for a ping message.

If the message is a response to an Application-initiated request, the Decision Engine service directs the flow to:

1. Choose the XREF cookie information.
2. Publish the notification to the waiting business process.
3. Insert the PeopleSoft time stamp.
4. Reply to PeopleSoft.

To use this service, create a rule in the BPML:

```
<rule name="PingRule">
  <condition>isPing="ping"</condition>
</rule>
```

Configuring the Decision Engine Service for PS Receive

To configure the Decision Engine service:

1. Create a BPML rule and set the condition (for example, isPing= “ping”).
2. Create a Decision Engine service configuration, setting the case reference and the associated activity. See *Creating a Service Configuration..*
3. Define the associated activities.

The following code is an example of Decision service BPML for the PS Receive business process:

```
<choice>
  <select>
    <case ref="PingRule" activity="RespondToPing" />
    <case ref="PingRule" negative="true" activity="RespondToMessage" />
  </select>

  <sequence name="RespondToPing">
    ...
  </sequence>

  <sequence name="RespondToMessage">
    ...
  </sequence>
</choice>
```

Request Response DXREF Service for PS Receive

The Request Response DXREF service, along with the Request Response XREF service, provides the link between the PS Send and Wait and the PS Receive business processes.

1. The DXREF service checks the database for an entry matching the XREF cookie selected by the Typing service.
2. If the DXREF service finds a matching record, it retrieves the business process ID belonging to the consuming business process from the entry. The business process ID is set in the process data and passed to the Produce service, where it is used to notify the consuming business process.

Configuring the DXREF Service for PS Receive

To configure the DXREF service, create a service configuration. See *Creating a Service Configuration*.

The following code is an example of the Request Response DXREF service BPML for the PS Receive business process:

```
<operation name="De-Cross Reference">
  <participant name="RequestResponseDXREFService"/>
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

Translation Service for PS Receive

The Translation service creates two types of responses to incoming messages:

If the Message Is a	Then the Business Process Uses the Translation Service to
Simple ping request from PeopleSoft	Send an acknowledgment back to the HTTP Server adapter so that it can notify PeopleSoft that the ping was received correctly.
PeopleSoft response to a previous business request	Send a business acknowledgment back to the HTTP Server so that it can notify PeopleSoft that the business message was received correctly.

Configuring the PS Receive Translation Service for Ping Reply

The Translation service uses the header information from the incoming PeopleSoft message and creates a PeopleSoft-accepted reply message. For this to happen, create a map that translates an incoming PeopleSoft message to the reply.

1. On the input side, define the ping message.
2. On the output side, define the ping reply message.
3. Map all common attributes and elements from the input side to the output side.
Common attributes include namespace, interface, and member type.
4. Hard code the opnum attribute to **1** (#opnum="1");).
5. Set the **return** field to "0" (#return="0");).
6. Check the map in to Application.
7. Create a PS Receive Translation service configuration. See *Creating a Service Configuration*. No BPML parameters are required.

Configuring the PS Receive Translation Service for Business Request Reply

To configure the PS Receive Translation service for business request reply:

1. Create a translation map with the input side defined by the ping message and the output side defined by the ping-reply message.
2. Map all common attributes and elements from the input side to the output side.
Common attributes include namespace, interface, and member type. Common elements include publishingnode, channel, and publicationid.
3. Hard code the opnum attribute to **1** (#opnum="1").
4. Set the type attribute to **number** (#type="number").
5. Set the **return** field to **0**(#return="0").
6. Check the map in to Application.
7. Create the PS Receive Translation service configuration. See *Creating a Service Configuration*. No BPML parameters are required.

Produce Service for PS Receive

The Consume and Produce services facilitate communication between the PS Send and Wait and PS Receive business processes. The link created by the Request Response XREF and Request Response DXREF services enables this communication. The Produce service persists the PeopleSoft message in the database. The Produce service notifies the consuming business process associated with the business process ID received from the DXREF service. The notified business process retrieves the message from the database and continues.

Configuring the Produce Service for PS Receive

To configure the Produce service for PS Receive:

1. Create a Produce service configuration. See *Creating a Service Configuration*.
2. Set the INVOKE_ID_LIST parameter, using the following convention:

```
"consuming business process name"/"text()"
```

The DXREF service sets the consuming business process ID in the process data using the consuming business process name. This parameter extracts the ID from the process data.

3. Set the output message attribute to the name of the Consume service input message.
4. Set a parameter to send the PeopleSoft message to the consuming business process. The consuming business process accesses this parameter to get the message. Assign this parameter from //PrimaryDocument.

The following code is an example of Produce service BPML for the PS Receive business process:

```
<produce>
  <participant name="ConsumeWF"/>
  <output message="anOutMsg">
    <assign to="INVOKE_ID_LIST" from="//PSSend_Wait/text()"/>
    <assign to="aDocument" from="//PrimaryDocument" append="true"/>
  </output>
</produce>
```


HTTP Respond Service for PS Receive

Use the HTTP Respond service to send a response to the Perimeter server, which responds to PeopleSoft. The HTTP Respond service notifies the Perimeter server that the response from PeopleSoft has been processed and that it can send an acknowledgment to PeopleSoft.

Configuring the HTTP Respond Service for PS Receive

To configure the HTTP Respond service for the PS Receive process, include the service configuration provided with Application in your business process, and set the doc-has-headers parameter to **False**.

The following code is an example of HTTP Respond service BPML for the PS Receive business process:

```
<operation name="HttpResponse">
  <participant name="HttpRespond" />
  <output message="Xout">
    <assign to="doc-has-headers">false</assign>
    <assign to=".">from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

Timestamp Service for PeopleSoft for PS Receive

The Timestamp Service for PeopleSoft provides two modifications for the reply to PeopleSoft. The service inserts:

1. An ISO-8061 formatted time stamp into the <publishtimestamp> element of the reply.
2. An XML processing instruction at the beginning of the reply document.

Configuring the Timestamp Service for PeopleSoft

To configure the Timestamp Service for PeopleSoft, create the service configuration. See *Creating a Service Configuration*.. No BPML parameters are required.

The following code is an example of PS Time Stamp service BPML for the PS Receive business process:

```
<operation name="Insert ISO-8061 formatted TimeStamp for PeopleSoft">
  <participant name="ExamplePeopleSoftTimeStampService" />
  <output message="Xout">
    <assign to="." from="*" />
  </output>
  <input message="Xin">
    <assign to="." from="*" />
  </input>
</operation>
```

Adapter for PeopleSoft CRM CIC

The Adapter for PeopleSoft CRM CIC (formerly the Vantive adapter) enables a Application business process to perform create, read, update, and delete operations on PeopleSoft CRM CIC system data.

The following table provides an overview of the Adapter for PeopleSoft CRM CIC:

System name	None
Graphical Process Modeler (GPM) category	None
Description	Allows the users to interact with their PeopleSoft CRM CIC systems from within their Application business processes. The business process may perform the following operations on their PeopleSoft CRM system data: create, read, update or delete.
Preconfigured?	No. There may be multiple configurations of the adapter. One configuration of Adapter for PeopleSoft CRM CIC corresponds to one PeopleSoft CRM CIC server. If there is more than one PeopleSoft CRM CIC server running on different host/port, more configurations are needed.
Requires third party files?	If using Translation service to convert XML data, must have a DTD.
Platform availability	<ul style="list-style-type: none">◆ Microsoft Windows◆ Sun Solaris◆ HP-UX◆ IBM-AIX
Related services	Translation service
Application requirements	Requires a PeopleSoft CRM CIC server set up with appropriate permission.
Invocation	Runs as part of a business process. See <i>Business Process Examples</i> on page 1100.
Business process context considerations	None
Returned status values	Basic status: Success or Failure Advanced Statuses: see <i>Error Messages</i> on page 1097.
Restrictions	Each adapter can have up to 20 simultaneous connections to the PeopleSoft CRM CIC server. This is to prevent the out of memory error on the server. If you set the Max. Connections parameter to a value greater than 20, the software will set it to 20.

Testing considerations

The best way to test the Adapter for PeopleSoft CRM CIC is to compose an XML file and pass it to the business process to conduct create, update, read and delete operations and verify the result is correct after each step.

Depending on the PeopleSoft CRM CIC system configuration, there might be some restrictions on some operations. If that is the case, the error message from the PeopleSoft CRM CIC system will be passed to the user in the output XML document, and the user can contact the PeopleSoft CRM CIC system administrator.

In case any error occurred, the error message is also logged in the log file.

Requirements

The Adapter for PeopleSoft CRM CIC has the following requirements and restrictions:

HP-UX Platform

- ◆ If your system has a C run-time library patch PHSS_22543 or PHSS_24627 installed, make sure you use the VanAPI version 9.0.2.2 or later.
- ◆ Application does not support creating related objects from the PS CRM adapter on the HP-UX platform.

AIX® Platform – The PeopleSoft CRM CIC server requires the following components for the Oracle 8.1.7 on AIX 4.3.3 platform: Vantive API needs C run-time libraries with level 5.0.0.0 or later.

Compatibility with Vantive 8 servers – The VanAPI library version 8.5.4.20 does not work with Application. Do not use this version's shared libraries on any platform. Use VanAPI version 8.5.4.11 only.

How the Adapter for PeopleSoft CRM CIC Works

Use the Adapter for PeopleSoft CRM CIC to read or write data to your PeopleSoft CRM CIC system. Your customer input XML document specifies the operation and the field or record on which you want to operate, and the Adapter for PeopleSoft CRM CIC translates this to PeopleSoft CRM CIC requests.

For example, you have customer information stored in PeopleSoft CRM CIC. To provide the marketing department with the customer address information from the PeopleSoft CRM CIC system, you can access the information within a business process in Application using the Adapter for PeopleSoft CRM CIC to read this information and return it in the form of an XML output file.

The following steps summarize how you might use the Adapter for PeopleSoft CRM CIC in a business process.

1. Application processes a file and passes it to the Adapter for PeopleSoft CRM CIC at run time.
2. The Adapter for PeopleSoft CRM CIC parses the file.
3. The Adapter for PeopleSoft CRM CIC performs the operation specified in the file on the PeopleSoft CRM CIC system.

4. The result is passed back to Application.
5. Application performs the next operation in the business process.

Note: Results from one environment may not be returned in the same order as the results from another environment.

Data Structure and the Adapter for PeopleSoft CRM CIC

The Adapter for PeopleSoft CRM CIC operates on data and records contained in objects on a PeopleSoft CRM CIC server. Do not use the adapter to change the underlying structure of those objects.

The Adapter for PeopleSoft CRM CIC operates on the logical view of the PeopleSoft CRM CIC data as supplied by the VanAPI-Java Technology Edition (JTE). The VanAPI library contains public C-style functions that establish a connection to the PeopleSoft CRM CIC server and manipulate the data in a PeopleSoft CRM CIC application database.

In the PeopleSoft CRM CIC system, data has the following structure:

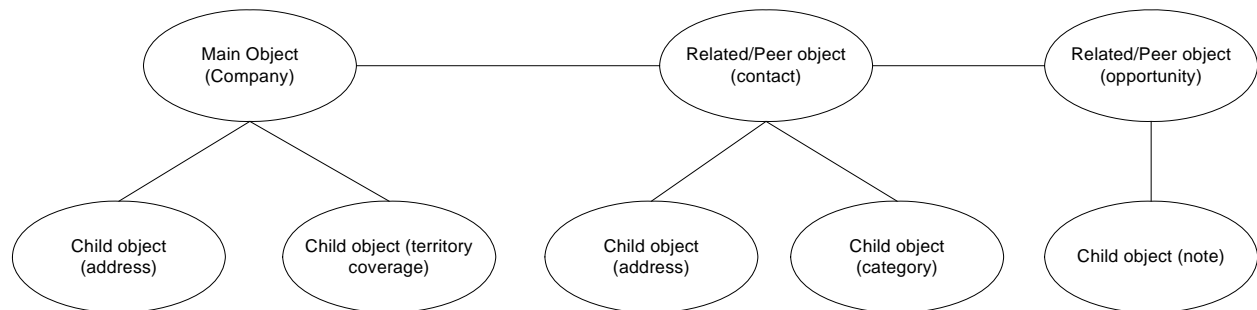
- Object hierarchy
- Object type
- Record set or record
- Field

Object Hierarchy

The PeopleSoft CRM CIC system organizes data in a hierarchy:

- Main object – Highest-level object. Can contain unlimited related (peer) objects and child objects.
- Related object – Can contain unlimited related objects and child objects. A related object has to belong to a main object or another related object. A related object can be a main object if accessed directly.
- Child object – Lowest-level object. Cannot contain related objects or child objects. A child object has to belong to either a main object or a related object. A child object cannot be a main object in another context.

The following figure shows an example of object relationships in the PeopleSoft CRM CIC client: One main object, company, contains one related object, contact. Company contains two child objects, address and territory coverage. In addition, contact contains one related object, opportunity, and two child objects, address and category. Opportunity contains one child object, note.



Object Type

An object type represents a real world entity or logical concept that is essential for a business organization. An object type can be an *account* or *company*. All records in the PeopleSoft CRM CIC system have an associated object type.

Record Set or Record

A record set contains instances of an object type. A record is accessed through a record set, and is one instance of an object type (main, related, or child). For example, an instance of the object type *company* is a record that could contain the data for Acme, Inc. The data contained by this record is in fields.

Field

A *field* is the lowest container of data in the hierarchy, and represents a single element. A record can contain one or more fields. For example, in the Acme, Inc. record, one field might contain the number of employees. The Adapter for PeopleSoft CRM CIC enables creating, reading, updating, or deleting values in the fields.

Adapter for PeopleSoft CRM CIC Operations

The operations supported by the Adapter for PeopleSoft CRM CIC are create, read, update, and delete (CRUD). To operate on an object type, you need to specify an object path. In the object path, specify all object types by their labels, which correspond to the labels in the PeopleSoft CRM CIC client interface. For example, if you want to update a customer address record, you must specify the *customer* object and the *address* object.

The object element maps to the model hierarchy. Element object.1 is always a main object, and the nested object elements can be either child or related objects. If an element represents a child object, then it cannot have an object element as its child.

The following tags are used in the Adapter for PeopleSoft CRM CIC XML:

Request Tag – Defines the operation being requested (create, read, update, delete). You can perform operations on multiple objects at one time, but they must have the same object type (for example, address).

Object Tag – Includes a label attribute, which is the display label in the PeopleSoft CRM CIC client GUI. This label is the name of any main, related, or child objects. An object can contain another object, which can contain another object, and so on. The highest-level object has to be a main object. Anything between the main object and the lowest-level object has to be a related object. The lowest-level object can either be a main, related, or child object.

Search Tag – Defines the field name on which to search; use the search element to represent the search criteria. Each search element is one search criterion. The field attribute is the display label of the column name in the table. The operator attribute is one of the following operators: =, <>, <, >, in, ^ in, btwn, like, sndx, pat. You can perform searches on any object type.

The value of the search element is the value to compare against. You can specify search criteria for each level of object and all of them must be satisfied to get the read results. The search criteria are implicitly tied together with the AND statement. The PeopleSoft CRM CIC API does not support OR.

Field Tag – Defines the fields to be created, searched (read), updated, or deleted. Field tags are included on only the lowest-level object.

XML tags that occur multiple times in a document must be unique if they require fixed data from the DTD. To make the tags unique, append a period (.) and a number after the tag.

Create

The create operation creates a record, which can be a main, related, or child object. The operation attribute of the request element must be *Create*. The create operation creates records on only one object type at a time. For example, in the following XML, although you can specify both Customer and Addresses, you can create records only on the lowest level object, in this case, Addresses. If you want to create a customer, which has customer addresses, you must perform separate operations for these object types.

The name attribute represents the display label of the columns to write data to. You can write more than one field from the lowest-level object in each create operation.

The following XML example shows a create operation for a Customer Address record. The Customer object has search criteria that specify on which customer object or objects the address record will be added. The Customer search criteria set the phone number to 799-4721. The address record that is created sets the following values:

Type is set to Ship To.

City is set to Aspen.

State/Province is set to CO.

```
<PS_CRMAadapter>
  <request operation="Create">
    <object.1 label="CUSTOMER">
      <search.1 field="Phone" operator="=">799-4721</search.1>
      <object.2 label="Addresses">
        <field.1 name="Type">Ship To</field.1>
        <field.2 name="City">Aspen</field.2>
        <field.3 name="State/Province">CO</field.3>
      </object.2>
    </object.1>
  </request>
</PS_CRMAadapter>
```

The response for the create operation does not have result records. The status attribute indicates whether the operation succeeded or failed. The statusText attributes provide the number of records created if the operation succeeded or a detailed error message if the operation failed.

The following XML shows a create response:

```
<PS_CRMAadapter>
  <results status="success" statusText="2 Record(s) Created."/>
</PS_CRMAadapter>
```

Read

The operation attribute of the request element must be *Read*. The read operation reads data from the field level and one object type at a time. For example, in the following XML example, you cannot read data for both Customer and Addresses; you can read data only for the lowest-level object, which is Addresses.

The name attribute represents the display label of the columns to read data from. You can read more than one field from the lowest-level object in each read operation.

The following XML example shows a read operation for the city and state/province information of the Mail To address from the customer whose phone number is 555-1234.

```
<PS_CRMAadapter>
  <request operation="Read">
    <object.1 label="CUSTOMER">
      <search.1 field="Phone" operator="=">555-1234</search.1>
      <object.2 label="Addresses">
        <search.2 field="Type" operator="=">Mail To</search.2>
        <field.1 name="City"/>
        <field.2 name="State/Province"/>
      </object.2>
    </object.1>
  </request>
</PS_CRMAadapter>
```

The following XML example shows the status attribute report for a successful read operation. Each of the result elements is a record returned from the read operation. The field elements are the same as those in the request XML, except the values are the data returned from the read operation.

```
<PS_CRMAadapter>
  <results status="success" statusText="2 Record(s) Read.">
    <result>
      <field.1 name="City">Dublin</field.1>
      <field.2 name="State/Province">OH</field.2>
    </result>
    <result>
      <field.1 name="City">Denver</field.1>
      <field.2 name="State/Province">CO</field.2>
    </result>
  </results>
</PS_CRMAadapter>
```

Update

The update operation enables you to modify fields in a record. The operation attribute of the request element must be *Update*. Provide search criteria to select the records on which you want to operate.

The update operation updates data for only one object type at a time. For example, in the following XML, although you can specify search criteria for both Customer and Addresses, you can update data on only the lowest-level object, which is Addresses.

The name attribute represents the display label of the columns to update data. The user can update more than one field from the lowest-level object in each update operation.

The following XML example shows an update operation for the city and state/province information of the Mail To address for the customer whose phone number is 555-1234 and who has an address type set to Mail To.

```
<PS_CRMAadapter>
  <request operation="Update">
    <object.1 label="CUSTOMER">
      <search.1 field="Phone" operator="=">555-1234</search.1>
      <object.2 label="Addresses">
        <search.2 field="Type" operator="=">Mail To</search.2>
        <field.1 name="City">Aspen</field.1>
        <field.2 name="State/Province">CO</field.2>
      </object.2>
    </object.1>
  </request>
</PS_CRMAadapter>
```

```
</object.2>
</object.1>
</request>
</PS_CRMAAdapter>
```

The response for the update operation does not have any result records and is the same as a response from the create and delete operation.

Delete

The delete operation enables you to delete PeopleSoft CRM CIC records. You provide the search criteria to select the records to delete. The operation attribute of the request element must be *Delete*.

The following XML example shows a delete operation for address records. Each address record is a child of a customer with the phone number 799-4721 and an address type set to Mail To. The address records are deleted in this operation.

```
<PS_CRMAAdapter>
  <request operation="Delete">
    <object.1 label="CUSTOMER">
      <search.1 field="Phone" operator="=">799-4721</search.1>
    <object.2 label="Addresses">
      <search.2 field="Type" operator="=">Mail To</search.2>
    </object.2>
  </object.1>
</request>
</PS_CRMAAdapter>
```

The response for the delete operation does not have any result records.

Building Adapter for PeopleSoft CRM CIC DTDs

You must create DTDs for your operations if you want to use the Translation service. There are five different types of DTDs, one for each operation (create, read, update, and delete) and one for the response document. Create a DTD for each object type you want to operate on, because only one object type can be operated on at a time. For example, you might have one DTD for customer (main object), but if you want to operate on customer address (child object), you need to create a separate DTD.

Input DTDs

A DTD is required for the Translation service to translate the Adapter for PeopleSoft CRM CIC input XML document to and from other documents.

The following XML example shows a DTD for reading fields on a related object. The related object type in this case is Customer Address. Read the DTD in the following way:

The request operation is fixed and set to Read.

The main object is fixed and set to CUSTOMER.

Search criteria are specified for Customer, including Name =.

The child object is fixed and set to ADDRESSES.

The fields that are retrieved are Street1 and Street2 on the address objects.


```

<?xml version='1.0' encoding='UTF-8'?>

<!ELEMENT PS_CRMAdapter (request)>
<!ELEMENT request (object.1)>
<!ATTLIST request operation CDATA #FIXED "Read">

<!ELEMENT object.1 (search.1, object.2)>
<!ATTLIST object.1 label CDATA #FIXED "CUSTOMER">

<!ELEMENT search.1 (#PCDATA)>
<!ATTLIST search.1 field CDATA #FIXED "Name"
                operator CDATA #FIXED "=">

<!ELEMENT object.2 (field.1, field.2?)>
<!ATTLIST object.2 label CDATA #FIXED "ADDRESSES">

<!ELEMENT field.1 EMPTY>
<!ATTLIST field.1 name CDATA #FIXED "Street1">

<!ELEMENT field.2 EMPTY>
<!ATTLIST field.2 name CDATA #FIXED "Street2">
Output DTDs
The results tag contains the results for the operation being performed. For more
information, see Error Messages

```

Error Messages

The PeopleSoft CRM CIC server returns an error status for the following conditions:

- Field – The supplied field value does not match the list of valid values for the specified field.
- Data type – The supplied field value has an invalid data type for the specified field.
- Object type/object path error – An invalid object type or object path is specified.
- Privileges – The user does not have appropriate permissions for the attempted operation.

The response XML returns a status message in the form of success or failure. In some cases, an advanced status message (called *statusText* in DTD) is returned. The following table describes the advanced status messages:

Message	Description
Request XML document is not valid	An error occurred when parsing the XML; the XML input document is structured incorrectly.
General adapter error	Something unexpected occurred in the Adapter for PeopleSoft CRM CIC.
Response XML error	An error was detected while the response XML was created.
Cannot get a connection to PeopleSoft CRM CIC server	<p>Connection to the PeopleSoft CRM CIC server was not completed for one of the following reasons:</p> <ul style="list-style-type: none"> ◆ The user name and password are already in use. ◆ The connection pool is full. ◆ An error occurred in the PeopleSoft CRM CIC server while trying to obtain the connection.
Failed to perform operation	System error.
No records found	No records were found in the read operation; the status is still Success.

Implementing the Adapter for PeopleSoft CRM CIC

To implement the Adapter for PeopleSoft CRM CIC for use in a business process:

1. Activate your license for the Adapter for PeopleSoft CRM CIC. See *An Overview of Implementing Services*.
2. Install the Adapter for PeopleSoft CRM CIC. See *Installing the Adapter for PeopleSoft CRM CIC* on page 1098.
3. Create an Adapter for PeopleSoft CRM CIC configuration. See *Creating a Service Configuration*.
4. Configure the Adapter for PeopleSoft CRM CIC. See *Configuring the Adapter for PeopleSoft CRM CIC for Application* on page 1099.

Installing the Adapter for PeopleSoft CRM CIC

To install the Adapter for PeopleSoft CRM CIC, run the `install3rdParty.sh` script to install the required PeopleSoft CRM CIC .jar file (`vanjavi.jar`) and shared libraries.

The Adapter for PeopleSoft CRM CIC requires operating system-specific shared libraries. The supported operating systems are Solaris™, HP-UX, and AIX. The following required library files are specified by platform:

Solaris – libvanjavi.so, libvanjavi_g.so, vanapi.so, vanconv.so, libvanres.so.1
 HP-UX – libvanjavi.sl, libvanjavi_g.sl, libvanres.sl, vanapi.h, vanconv.h, vanapi.sl
 AIX – libvanjavi.so, libvanjavi_g.so, libvanres.a, vanapi.so, vanconv.a

Contact Sterling Commerce Customer Support for a custom installation of the Adapter for PeopleSoft CRM CIC for your platform version.

Pass the absolute path for the library files to the install3rdParty.sh script.

For the PeopleSoft CRM CIC .jar files, run the following script from the folder where you have Application installed:

```
Vendor Version location of the .jar
```

The following figure shows an example of the script for the PeopleSoft CRM CIC .jar files:

For the PeopleSoft CRM CIC shared libraries, run:

```
Vendor Version location of the lib
```

The following figure shows an example of the script for the PeopleSoft CRM CIC shared libraries:

Configuring the Adapter for PeopleSoft CRM CIC for Application

The following table describes the fields used to configure the Adapter for PeopleSoft CRM CIC in Application:

Caution: The values for PeopleSoft CRM CIC Host, PeopleSoft CRM CIC Port, Default Login, Default Password, Idle timeout, Connection wait timeout, and Max. connections are set up in the service configuration, and *cannot* be overridden in the business process parameters.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
PeopleSoft CRM CIC Host	Host name or IP address of the computer where the PeopleSoft CRM CIC system is running. Do not specify the PeopleSoft CRM CIC host name in the business process. Required.

Field	Description
PeopleSoft CRM CIC Port	IP port number on the host that is listening for requests. Do not specify the port number in the business process. Required.
Default Login	Default login name for authentication with the PeopleSoft CRM CIC system. This login name is used if the business process does not specify login parameters. Optional.
Default Password	Default password for authentication with the PeopleSoft CRM CIC system. The default password is used if the business process does not specify login parameters. Optional.
Idle timeout (sec)	Amount of time in seconds to allow a PeopleSoft CRM CIC connection to idle before it is closed. The default value is 600. Required.
Connection wait timeout (sec)	Amount of time in seconds to wait for a PeopleSoft CRM CIC connection to become available before sending an error message. The default value is 60. Required.
Max. Connections	Maximum number of connections to PeopleSoft CRM CIC that the adapter is allowed to have open at one time. The default value is 5. Because of the amount of memory consumed by a connection, there is an upper limit of 20. If a value larger than 20 is entered, the Adapter for PeopleSoft CRM CIC converts the value to 20 without sending an error message. Required.

Connection Management

The PeopleSoft CRM CIC system cannot handle multiple users with the same login parameters accessing PeopleSoft CRM CIC simultaneously. The Adapter for PeopleSoft CRM CIC uses a connection management component that operates on the premise that each Adapter for PeopleSoft CRM CIC should connect to a different PeopleSoft CRM CIC system. If a connection to PeopleSoft CRM CIC is in use for the specified login name and password combination, a new request is delayed until the business process using the existing connection is finished. The Adapter for PeopleSoft CRM CIC returns an error if the business process has not finished using the connection before the connection wait timeout period has expired.

The connection management component supports two types of PeopleSoft CRM CIC connections:

Default – Used if the business process does not specify login parameters. If the business process does not specify login parameters, the configuration login parameters are used. An error occurs if no login parameters are specified in either the business process or the adapter configuration.

Trusted connections – Used if login parameters are specified in the business process.

Both trusted and default connections are opened when requested, if not already open. An open connection is closed after the idle timeout period has expired, or when the connection is idle and a new connection needs to be opened.

Business Process Examples

The Adapter for PeopleSoft CRM CIC can be invoked from a business process, as shown in the following example:

You can also pass in values for login and password to overwrite the default login and password to conduct operations that require more privileges. Following is an example of a business process document with user specified login and password:

Timestamp Service for PeopleSoft

The following table provides an overview of the Timestamp Service for PeopleSoft:

System name	None
Graphical Process Modeler (GPM) categories	All Services, Applications > ERP
Description	Inserts a timestamp in a PeopleSoft-specific format within acknowledgements sent to PeopleSoft for responses received. You can use this service only with the Adapter for PeopleSoft.
Business usage	You must use the Timestamp Service for PeopleSoft with the Adapter for PeopleSoft, because it inserts the timestamp into an acknowledgement being sent back to PeopleSoft for a response.
Preconfigured?	No
Requires third party files?	None
Platform availability	All supported Application platforms
Related services	Adapter for PeopleSoft
Application requirements	You must create a configuration of the Timestamp Service for PeopleSoft for use inside a BPML service definition. No configuration or BPML parameters are required.
Initiates business processes?	None
Invocation	Invoke the Timestamp Service for PeopleSoft from a business process as shown in the following example: <pre><operation name="Insert TimeStamp"> <participant name="PeopleSoftTimeStamp"/> <output message="Xout"> <assign to="." from="**"/> </output> <input message="Xin"> <assign to="." from="**"/> </input> </operation></pre>
Business process context considerations	The Timestamp Service for PeopleSoft is instantiated at runtime by the business process engine, so you can define a single instance for use by multiple business processes at the same time (as the business process engine instantiates a new object for each business process in which they are invoked).
Returned status values	Success, Failure
Restrictions	You can use the Timestamp Service for PeopleSoft only with the Adapter for PeopleSoft.

Testing considerations Use the Timestamp Service for PeopleSoft inside a PS Receive business process where a non-ping message is received from the PeopleSoft system.

PGP Package Service (Build 4300 - Build 4314)

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Package service, in conjunction with the PGP Profile Manager, enables you to encrypt and digitally sign documents using PGP.

The following table provides an overview of the PGP Package service:

System name	PGP Package service
Graphical Process Modeler (GPM) category	All Services
Description	This service encrypts and digitally signs a document based on the Open PGP standard, using public key or conventional cryptography.
Business usage	Use this service to encrypt and sign a document in the document area of process data.
Usage example	A business process is executed to encrypt and sign a document, based on the information stored in a PGP profile.
Preconfigured?	Yes. A configuration called PGP Package Service is installed with Application.
Requires third-party files?	No

Platform availability	<p>All supported Application platforms, with the following restrictions:</p> <p>For NAI McAfee eBusiness Server 8.1:</p> <ul style="list-style-type: none"> ◆ IBM AIX 4.2 or later ◆ HP-UX 10.20 or later ◆ Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) ◆ SuSE Linux for IBM S/390 and IBM Zseries <p>For NAI McAfee eBusiness Server 8.5</p> <ul style="list-style-type: none"> ◆ Solaris 9 or later <p>For NAI McAfee eBusiness Server 8.5.1</p> <ul style="list-style-type: none"> ◆ Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) ◆ Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) ◆ Microsoft Windows Server 2003 ◆ Microsoft Windows XP Professional Version 2002 Service Pack 2 <p>For Massachusetts Institute of Technology (MIT) Command Line Freeware</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 ◆ UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later <p>For PGP Corporation PGP® Command Line 9.5</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 SP1 Microsoft Windows 2000 (SP 4) ◆ UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 ◆ Mac OS X 10.4 or greater
Related adapters and services	<p>The PGP Package service works with the following services:</p> <ul style="list-style-type: none"> ◆ Command Line Adapter 2 ◆ PGP Unpackage service

Application requirements	<p>Before using this service, install one of the following:</p> <ul style="list-style-type: none"> ◆ McAfee E-Business Server (version 8.1, 8.5, or 8.5.1) from Network Associates Technology, Inc. ◆ PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) ◆ PGP Command Line (version 9.5) from PGP Corporation <p>Note: Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.</p>
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	<p>Basic statuses are:</p> <ul style="list-style-type: none"> ◆ 0 - Success ◆ 1- Error <p>See <i>Advanced Status Messages</i> on page 1114 for a list of advanced statuses. Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrCode].</p>
Restrictions	None
Persistence level	None
Testing considerations	<p>Create the profile in the PGP Profile Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.</p> <p>A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.</p> <p>To start the remote adapter implementation of the command line adapter:</p> <ol style="list-style-type: none"> 1. Locate the client jar (CLA2Client.jar in <i>Install_DIR</i>>/<client>/<cmdline2>) that contains all the necessary classes. 2. Move the client jar to the machine that has the PGP server installed. 3. Start the remote adapter implementation using the following command: <pre>java -jar CLA2Client.jar <port> [debug]</pre> <p>For example:</p> <pre>java -jar CLA2Client.jar 15699 debug</pre> <p>Note: The [debug] option is not required.</p>

Implementing the PGP Package Service

To implement the PGP Package service, complete the following tasks:

1. Activate your license for the PGP Package service. See *Managing Services and Adapters*.
2. Create a PGP profile, using the Application PGP Profile Manager. See *PGP Profile Manager*.
3. Create a PGP Package service configuration. See *Managing Services and Adapters*.
4. Configure the service. See *Configuring the PGP Package Service* on page 1107.
5. Use the PGP Package service in a business process.

Configuring the PGP Package Service

Before configuring, consider the following:

`public_user` (if using Public Key Cryptography) or `conv_keymap_name` (if using Conventional Cryptography) must be present for PGP Package service to perform encryption.

`secret_keymap_name` must be present for PGP Package service to perform signing.

To perform encryption and signing, a combination of both the previous statements applies.

If `public_user` and `conv_keymap_name` appear in the same business process, public key encryption will take precedence.

To configure the PGP Package service, specify settings specify the settings for the fields in the GPM. These fields are described in the following table:

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files used for encryption and signing will be read from or written to. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
profile_name	Name of PGP profile from the PGP Profile Manager. Required.
compress	Compression to be done before encryption or signing. Valid value is On. Default is On. Required for encryption and signing.
public_user	User name or key ID in the public key ring. Required for encryption (public key cryptography).
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for signing (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for encryption (conventional cryptography).
conv_cipher	The symmetric cipher to use when performing a conventional encryption operation (that is, <code>conv_keymap_name</code> is used). Valid values are: IDEA, CAST5, 3DES, AES128, AES196, AES256, Twofish. Default is IDEA. Optional.

Field	Description
DocumentId	The document identifier referenced to the document to be processed specifically. The default document for processing is the primary document. Optional.
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
ascii_armor	Whether to encode the file with McAfee E-Business Server's base-64 encoding (ASCII-armored format). Valid values are On and Off. Default is On. Optional.
textmode	Whether the input data is ASCII text and should be converted to canonical new lines before encryption. Valid values are On and Off. Default is Off. Optional.
outputfilename	<p>Output file name.</p> <p>For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc.</p> <p>For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of the document and is appended with the following:</p> <ul style="list-style-type: none"> ◆ *.asc during normal encryption ◆ .exe during sda process ◆ .pga during pgparchive process <p>Optional.</p>
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.
clearsig	Generates a signed message that can be read without PGP. The recipient must still use PGP to verify the signature. Unencrypted PGP-signed messages have a signature certificate pre-pended in binary form. The signed message is compressed. Therefore, it is unreadable by humans even though it is not encrypted. Cannot be used with EncryptAndSign on the command line. If you enable clearsig, it is recommended you enable ascii_armor and textmode also. Valid values are On and Off. Default is Off. Optional.
info	<p>How much information is returned. Valid values are:</p> <ul style="list-style-type: none"> ◆ Quiet - Only displays error messages. Not applicable to PGP Command Line. If selected defaults to normal mode. ◆ Normal - Displays warnings and error messages. Default. ◆ Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). If selected with other versions, defaults to normal mode. ◆ Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line. If selected, defaults to normal mode. <p>Optional.</p>

Field	Description
sda	<p>Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified.</p> <p>Creates a self-decrypting executable file, which is conventionally encrypted using a passphrase. The resulting file can be decrypted by double-clicking it and entering the passphrase. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed.</p> <p>SDA files can be created with any platform that McAfee E-Business Server (version 8.1 or later) supports, but can be executed only on Windows platforms.</p> <p>To create sda files with PGP Command Line (version 9.5), set the target_platform parameter (described later in this table).</p> <p>The default file extension is .exe.</p> <p>Note: The sda file cannot exceed 4 GB after compression.</p> <p>Valid values are On and Off. Default is Off. Optional.</p>
pgparchive	<p>Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified.</p> <p>Creates a file that can be decrypted using the archive reader, which can be redistributed freely. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed.</p> <p>The default extension is .pga.</p> <p>Valid values are On and Off. Default is Off. Optional.</p>
discard_paths	<p>Applicable only with sda or pgparchive. Strips relative path information from the list of files in a sda or pgparchive. During the decryption of the archive, the files are placed in the current directory instead of in subdirectories of the current directory. Optional.</p>
target_platform	<p>Applicable only with PGP Command Line (version 9.5) and sda. Specifies the platform an sda file can be decrypted on. Valid values are:</p> <ul style="list-style-type: none"> ◆ win32 ◆ linux ◆ solaris ◆ aix ◆ hpux ◆ osx <p>Default is the current platform. Optional.</p>

Parameters Passed from Service to BP

The following table contains the parameters that are passed from the PGP Package service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are: <ul style="list-style-type: none"> ◆ ENCRYPT ◆ ENCRYPT_SIGN ◆ SIGN Required.
FileName (PGP/FileName)	Name of the file being processed. Required.
Document (PGP/Document)	The processed document is placed in Process Data – not as Primary Document. The attribute is the SCIObjctID, which enables a hyperlink for viewing the content of the processed document. Required.
DocumentId (PGP/DocumentId)	Document identifier of the document. Required.
Status (PGP/Status)	Process status. Valid values are Success and Error. Required.
ErrorCode (PGP/ErrorCode)	Value returned from executing PGP commands. Displayed when the Status is Error. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. Displayed when the Status is Error. Optional.

Business Process Example - Encrypt Operation (Public Key Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area. The profile is based on PGP107. In this example, you use the default Command Line2 adapter configuration, PGPCmdlineService, to execute the encrypt command. You want to use the working directory, remote name and port stated in the BPML. Therefore, these values override the pre-configured values in PGPCmdLineService. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption.

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Encrypt Operation (Conventional Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area of process data. The profile is based on PGP107. In this example, you use the Command Line2 adapter configuration, MyCLA2, to execute the commands. The remote name, port, and working directory are pre-configured in the service configuration. The value of conv_keymap_name, Conv_abc_tp, which must be in the profile's conventional key map, is used for conventional encryption:

```

<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="conv_keymap_name">Conv_abc_tp</assign>
        <assign to="conv_cipher">CAST5</assign>
        <assign to="cmdline2svcname">MyCLA2</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - Encrypt and Sign Operation (Public Key Encryption)

The following business process uses the PGP Package service to encrypt and sign the primary document in the document area. For signing, you need to pass in the secret_keymap_name, which must be in the PGP107 profile's secret key map. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption. In this example, you choose not to compress the document before signing and encryption.

```

<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">off</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
        <assign to="secret_keymap_name">my_secret</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

</output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</process>

```

Business Process Example - Encrypt and Sign Operation (Conventional Encryption)

The following business process uses PGP Package Service to encrypt and sign the Primary Document in the document area. For signing, the user needs to pass in the secret_keymap_name, which must be present in the PGP107 profile's Secret Key Map. The value of conv_keymap_name, Conv_abc_tp, which must be present in the Profile's Conventional Key Map, is used for conventional encryption. The user chooses not to compress the document before signing and encryption.

```

<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">off</assign>
        <assign to="workingDir">/localsvr/share/tmp</assign>
        <assign to="remoteName">nn.nnn.nn.nnn</assign>
        <assign to="remotePort">xxxxx</assign>
        <assign to="conv_keymap_name">Conv_abc_tp</assign>
        <assign to="conv_cipher">CAST5</assign>
        <assign to="secret_keymap_name">si_secret</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - Encrypt Operation (Public Key Encryption) Using a Specific Document ID

The following business process uses the PGP Package service to encrypt a document, with the document ID columbia:1774b9b:feaea8ae12:-6ea8 in the document area.

```

<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One"> PGPPackageService
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
      </output>
    </operation>
  </sequence>
</process>

```



```

        <assign to="public_user">0x2343</assign>
        <assign to="DocumentId">columbia:1774b9b:feaea8ae12:-6ea8</assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Sign Operation

The following business process uses the PGP Package service to sign the primary document in the document area.

```

<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
        <assign to="secret_keymap_name">my_secret</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - OnFault Handling

The following business process shows the onFault handling for the PGP Package service.

```

<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/localsvr/share/tmp</assign>
        <assign to="remoteName">nn.nnn.nn.nnn</assign>
        <assign to="remotePort">12345</assign>
        <assign to="secret_keymap_name">si_secret</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

```

    <assign to="Status">The file is signed successfully</assign>
  <onFault>
    <assign to="Status">General Error Occurred</assign>
  </onFault>
  <onFault code="[PGPErrorCode] Signature Check error">
    <assign to="Status">Incorrect signature</assign>
  </onFault>
</sequence>
</process>

```

Advanced Status Messages

The following table contains exit codes from the McAfee E-Business Server and PGP Command Line Freeware. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.
160	Complete failure during a file wipe.
161	Partial fail, partial success during a file wipe (one file wiped, one not, for example).
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

The following table contains errors that result from the PGP Package service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId	The DocumentId value given in the BPML is incorrect.
Fail to get data from Primary Document.: There is no Primary Document	Primary Document is mandatory.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Profile Manager	The profile_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap	The secret_keymap_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap	The conv_keymap_name value given in the BPML is incorrect.

PGP Package Service (Build 4315 - Build 4321)

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Package service, in conjunction with the PGP Profile Manager, enables you to encrypt and digitally sign documents using PGP.

The following table provides an overview of the PGP Package service:

System name	PGP Package service
Graphical Process Modeler (GPM) category	All Services
Description	This service encrypts and digitally signs a document based on the Open PGP standard, using public key or conventional cryptography.
Business usage	Use this service to encrypt and sign a document in the document area of process data.
Usage example	A business process is executed to encrypt and sign a document, based on the information stored in a PGP profile.
Preconfigured?	Yes. A configuration called PGP Package Service is installed with Application.
Requires third-party files?	No

Platform availability	<p>All supported Application platforms, with the following restrictions:</p> <p>For NAI McAfee eBusiness Server 8.1:</p> <ul style="list-style-type: none"> ◆ IBM AIX 4.2 or later ◆ HP-UX 10.20 or later ◆ Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) ◆ SuSE Linux for IBM S/390 and IBM Zseries <p>For NAI McAfee eBusiness Server 8.5</p> <ul style="list-style-type: none"> ◆ Solaris 9 or later <p>For NAI McAfee eBusiness Server 8.5.1</p> <ul style="list-style-type: none"> ◆ Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) ◆ Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) ◆ Microsoft Windows Server 2003 ◆ Microsoft Windows XP Professional Version 2002 Service Pack 2 <p>For Massachusetts Institute of Technology (MIT) Command Line Freeware</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 ◆ UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later <p>For PGP Corporation PGP® Command Line 9.5</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 SP1 Microsoft Windows 2000 (SP 4) ◆ UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 ◆ Mac OS X 10.4 or greater
Related adapters and services	<p>The PGP Package service works with the following services:</p> <ul style="list-style-type: none"> ◆ Command Line Adapter 2 ◆ PGP Unpackage service

Application requirements	<p>Before using this service, install one of the following:</p> <ul style="list-style-type: none"> ◆ McAfee E-Business Server (version 8.1, 8.5, or 8.5.1) from Network Associates Technology, Inc. ◆ PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) ◆ PGP Command Line (version 9.5) from PGP Corporation <p>Note: Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.</p>
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	<p>Basic statuses are:</p> <ul style="list-style-type: none"> ◆ 0 - Success ◆ 1- Error <p>See <i>Advanced Status Messages</i> on page 1127 for a list of advanced statuses.</p> <p>Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrCode].</p>
Restrictions	None
Persistence level	None
Testing considerations	<p>Create the profile in the PGP Profile Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.</p> <p>A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.</p> <p>To start the remote adapter implementation of the command line adapter:</p> <ol style="list-style-type: none"> 1. Locate the client jar (CLA2Client.jar in <i>Install_DIR</i>/<i><client></i>/<i><cmdline2></i>) that contains all the necessary classes. 2. Move the client jar to the machine that has the PGP server installed. 3. Start the remote adapter implementation using the following command: <pre>java -jar CLA2Client.jar <port> [debug]</pre> <p>For example:</p> <pre>java -jar CLA2Client.jar 15699 debug</pre> <p>Note: The [debug] option is not required.</p>

Implementing the PGP Package Service

To implement the PGP Package service, complete the following tasks:

1. Activate your license for the PGP Package service. See *Managing Services and Adapters*.
2. Create a PGP profile, using the Application PGP Profile Manager. See *PGP Profile Manager*.
3. Create a PGP Package service configuration. See *Managing Services and Adapters*.
4. Configure the service. See *Configuring the PGP Package Service* on page 1120.
5. Use the PGP Package service in a business process.

Configuring the PGP Package Service

Before configuring, consider the following:

`public_user` (if using Public Key Cryptography) or `conv_keymap_name` (if using Conventional Cryptography) must be present for PGP Package service to perform encryption.

`secret_keymap_name` must be present for PGP Package service to perform signing.

To perform encryption and signing, a combination of both the previous statements applies.

If `public_user` and `conv_keymap_name` appear in the same business process, public key encryption will take precedence.

To configure the PGP Package service, specify settings specify the settings for the fields in the GPM. These fields are described in the following table:

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files used for encryption and signing will be read from or written to. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
profile_name	Name of PGP profile from the PGP Profile Manager. Required.
compress	Compression to be done before encryption or signing. Valid value is On. Default is On. Required for encryption and signing.
public_user	User name or key ID in the public key ring. Required for encryption (public key cryptography).
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for signing (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for encryption (conventional cryptography).
conv_cipher	The symmetric cipher to use when performing a conventional encryption operation (that is, <code>conv_keymap_name</code> is used). Valid values are: IDEA, CAST5, 3DES, AES128, AES196, AES256, Twofish. Default is IDEA. Optional.

Field	Description
DocumentId	The document identifier referenced to the document to be processed specifically. The default document for processing is the primary document. Optional.
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
ascii_armor	Whether to encode the file with McAfee E-Business Server's base-64 encoding (ASCII-armored format). Valid values are On and Off. Default is On. Optional.
textmode	Whether the input data is ASCII text and should be converted to canonical new lines before encryption. Valid values are On and Off. Default is Off. Optional.
outputfilename	Output file name. For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc. For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of the document and is appended with the following: <ul style="list-style-type: none"> ◆ *.asc during normal encryption ◆ .exe during sda process ◆ .pga during pgparchive process Optional.
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.
clearsig	Generates a signed message that can be read without PGP. The recipient must still use PGP to verify the signature. Unencrypted PGP-signed messages have a signature certificate pre-pended in binary form. The signed message is compressed. Therefore, it is unreadable by humans even though it is not encrypted. Cannot be used with EncryptAndSign on the command line. If you enable clearsig, it is recommended you enable ascii_armor and textmode also. Valid values are On and Off. Default is Off. Optional.
info	How much information is returned. Valid values are: <ul style="list-style-type: none"> ◆ Quiet - Only displays error messages. Not applicable to PGP Command Line. If selected defaults to normal mode. ◆ Normal - Displays warnings and error messages. Default. ◆ Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). If selected with other versions, defaults to normal mode. ◆ Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line. If selected, defaults to normal mode. Optional.

Field	Description
sda	<p>Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified.</p> <p>Creates a self-decrypting executable file, which is conventionally encrypted using a passphrase. The resulting file can be decrypted by double-clicking it and entering the passphrase. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed.</p> <p>SDA files can be created with any platform that McAfee E-Business Server (version 8.1 or later) supports, but can be executed only on Windows platforms.</p> <p>To create sda files with PGP Command Line (version 9.5), set the target_platform parameter (described later in this table).</p> <p>The default file extension is .exe.</p> <p>Note: The sda file cannot exceed 4 GB after compression.</p> <p>Valid values are On and Off. Default is Off. Optional.</p>
pgparchive	<p>Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified.</p> <p>Creates a file that can be decrypted using the archive reader, which can be redistributed freely. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed.</p> <p>The default extension is .pga.</p> <p>Valid values are On and Off. Default is Off. Optional.</p>
discard_paths	<p>Applicable only with sda or pgparchive. Strips relative path information from the list of files in a sda or pgparchive. During the decryption of the archive, the files are placed in the current directory instead of in subdirectories of the current directory.</p> <p>Optional.</p>
target_platform	<p>Applicable only with PGP Command Line (version 9.5) and sda. Specifies the platform an sda file can be decrypted on. Valid values are:</p> <ul style="list-style-type: none"> ◆ win32 ◆ linux ◆ solaris ◆ aix ◆ hpux ◆ osx <p>Default is the current platform. Optional.</p>

Parameters Passed from Service to BP

The following table contains the parameters that are passed from the PGP Package service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are: <ul style="list-style-type: none"> ◆ ENCRYPT ◆ ENCRYPT_SIGN ◆ SIGN Required.
FileName (PGP/FileName)	Name of the file being processed. Required.
inputFileNamePkg (PGP/inputFileNamePkg)	Name of the file contained in the PGP package. Optional.
Document (PGP/Document)	The processed document is placed in Process Data – not as Primary Document. The attribute is the SCIOBJECTID, which enables a hyperlink for viewing the content of the processed document. Required.
DocumentId (PGP/DocumentId)	Document identifier of the document. Required.
Status (PGP/Status)	Process status. Valid values are Success and Error. Required.
ErrorCode (PGP/ErrorCode)	Value returned from executing PGP commands. Displayed when the Status is Error. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. Displayed when the Status is Error. Optional.

Business Process Example - Encrypt Operation (Public Key Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area. The profile is based on PGP107. In this example, you use the default Command Line2 adapter configuration, PGPCmdlineService, to execute the encrypt command. You want to use the working directory, remote name and port stated in the BPML. Therefore, these values override the pre-configured values in PGPCmdLineService. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption.

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
      </output message="Xout">
    </operation name="One">
  </sequence name="optional">
</process name="PGP_Encrypt ">
```

```

        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Encrypt Operation (Conventional Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area of process data. The profile is based on PGP107. In this example, you use the Command Line2 adapter configuration, MyCLA2, to execute the commands. The remote name, port, and working directory are pre-configured in the service configuration. The value of conv_keymap_name, Conv_abc_tp, which must be in the profile's conventional key map, is used for conventional encryption:

```

<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="conv_keymap_name">Conv_abc_tp</assign>
        <assign to="conv_cipher">CAST5</assign>
        <assign to="cmdline2svcname">MyCLA2</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - Encrypt and Sign Operation (Public Key Encryption)

The following business process uses the PGP Package service to encrypt and sign the primary document in the document area. For signing, you need to pass in the secret_keymap_name, which must be in the PGP107 profile's secret key map. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption. In this example, you choose not to compress the document before signing and encryption.

```

<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">off</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
        <assign to="secret_keymap_name">my_secret</assign>
</output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Encrypt and Sign Operation (Conventional Encryption)

The following business process uses PGP Package Service to encrypt and sign the Primary Document in the document area. For signing, the user needs to pass in the secret_keymap_name, which must be present in the PGP107 profile's Secret Key Map. The value of conv_keymap_name, Conv_abc_tp, which must be present in the Profile's Conventional Key Map, is used for conventional encryption. The user chooses not to compress the document before signing and encryption.

```

<process name="PGP_Encrypt_Sign">
    <sequence name="optional">
        <operation name="One">
            <participant name=" PGPPackageService "/>
            <output message="Xout">
                <assign to="profile_name">PGP107</assign>
                <assign to="compress">off</assign>
                <assign to="workingDir">/localsvr/share/tmp</assign>
                <assign to="remoteName">nn.nnn.nn.nnn</assign>
                <assign to="remotePort">xxxxx</assign>
                <assign to="conv_keymap_name">Conv_abc_tp</assign>
                <assign to="conv_cipher">CAST5</assign>
                <assign to="secret_keymap_name">si_secret</assign>
                <assign to="." from="*"></assign>
            </output>
            <input message="Xin">
                <assign to="." from="*"></assign>
            </input>
        </operation>
    </sequence>
</process>

```

Business Process Example - Encrypt Operation (Public Key Encryption) Using a Specific Document ID

The following business process uses the PGP Package service to encrypt a document, with the document ID columbia:1774b9b:feaea8ae12:-6ea8 in the document area.

```

<process name="PGP_Encrypt ">
    <sequence name="optional">
        <operation name="One"> PGPPackageService
            <participant name="PGPPackageService"/>
            <output message="Xout">
                <assign to="." from="*"></assign>
                <assign to="profile_name">PGP107</assign>
                <assign to="compress">on</assign>
            </output>
        </operation>
    </sequence>
</process>

```

```

        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
        <assign to="DocumentId">columbia:1774b9b:feaea8ae12:-6ea8</assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Sign Operation

The following business process uses the PGP Package service to sign the primary document in the document area.

```

<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
        <assign to="secret_keymap_name">my_secret</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - OnFault Handling

The following business process shows the onFault handling for the PGP Package service.

```

<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/localsvr/share/tmp</assign>
        <assign to="remoteName">nn.nnn.nn.nnn</assign>
        <assign to="remotePort">12345</assign>
        <assign to="secret_keymap_name">si_secret</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">

```

```

    <assign to="." from="*"></assign>
  </input>
</operation>
  <assign to="Status">The file is signed successfully</assign>
<onFault>
  <assign to="Status">General Error Occurred</assign>
</onFault>
<onFault code="[PGPErrorCode] Signature Check error">
  <assign to="Status">Incorrect signature</assign>
</onFault>
</sequence>
</process>

```

Advanced Status Messages

The following table contains exit codes from the McAfee E-Business Server and PGP Command Line Freeware. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.
160	Complete failure during a file wipe.
161	Partial fail, partial success during a file wipe (one file wiped, one not, for example).
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

The following table contains errors that result from the PGP Package service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId	The DocumentId value given in the BPML is incorrect.
Fail to get data from Primary Document.: There is no Primary Document	Primary Document is mandatory.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Profile Manager	The profile_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap	The secret_keymap_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap	The conv_keymap_name value given in the BPML is incorrect.

PGP Package Service (Build 4322 or higher)

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Package service, in conjunction with the PGP Profile Manager, enables you to encrypt and digitally sign documents using PGP.

The following table provides an overview of the PGP Package service:

System name	PGP Package service
Graphical Process Modeler (GPM) category	All Services
Description	This service encrypts and digitally signs a document based on the Open PGP standard, using public key or conventional cryptography.
Business usage	Use this service to encrypt and sign a document in the document area of process data.
Usage example	A business process is executed to encrypt and sign a document, based on the information stored in a PGP profile.
Preconfigured?	Yes. A configuration called PGP Package Service is installed with Application.
Requires third-party files?	No

Platform availability	<p>All supported Application platforms, with the following restrictions:</p> <p>For NAI McAfee eBusiness Server 8.1:</p> <ul style="list-style-type: none"> ◆ IBM AIX 4.2 or later ◆ HP-UX 10.20 or later ◆ Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) ◆ SuSE Linux for IBM S/390 and IBM Zseries <p>For NAI McAfee eBusiness Server 8.5</p> <ul style="list-style-type: none"> ◆ Solaris 9 or later <p>For NAI McAfee eBusiness Server 8.5.1</p> <ul style="list-style-type: none"> ◆ Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) ◆ Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) ◆ Microsoft Windows Server 2003 ◆ Microsoft Windows XP Professional Version 2002 Service Pack 2 <p>For Massachusetts Institute of Technology (MIT) Command Line Freeware</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 ◆ UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later <p>For PGP Corporation PGP® Command Line 9.5</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 SP1 Microsoft Windows 2000 (SP 4) ◆ UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 ◆ Mac OS X 10.4 or greater
Related adapters and services	<p>The PGP Package service works with the following services:</p> <ul style="list-style-type: none"> ◆ Command Line Adapter 2 ◆ PGP Unpackage service

Application requirements	<p>Before using this service, install one of the following:</p> <ul style="list-style-type: none"> ◆ McAfee E-Business Server (version 8.1, 8.5, 8.5.1, or 8.6) from Network Associates Technology, Inc. ◆ PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) ◆ PGP Command Line (version 9.5) from PGP Corporation <p>Note: Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.</p>
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	<p>Basic statuses are:</p> <ul style="list-style-type: none"> ◆ 0 - Success ◆ 1- Error <p>See <i>Advanced Status Messages</i> on page 1140 for a list of advanced statuses.</p> <p>Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrCode].</p>
Restrictions	None
Persistence level	None
Testing considerations	<p>Create the profile in the PGP Profile Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.</p> <p>A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.</p> <p>To start the remote adapter implementation of the command line adapter:</p> <ol style="list-style-type: none"> 1. Locate the client jar (CLA2Client.jar in <i>Install_DIR</i>/<i><client></i>/<i><cmdline2></i>) that contains all the necessary classes. 2. Move the client jar to the machine that has the PGP server installed. 3. Start the remote adapter implementation using the following command: <pre>java -jar CLA2Client.jar <port> [debug]</pre> <p>For example:</p> <pre>java -jar CLA2Client.jar 15699 debug</pre> <p>Note: The [debug] option is not required.</p>

Implementing the PGP Package Service

To implement the PGP Package service, complete the following tasks:

1. Activate your license for the PGP Package service. See *Managing Services and Adapters*.
2. Create a PGP profile, using the Application PGP Profile Manager. See *PGP Profile Manager*.
3. Create a PGP Package service configuration. See *Managing Services and Adapters*.
4. Configure the service. See *Configuring the PGP Package Service* on page 1133.
5. Use the PGP Package service in a business process.

Configuring the PGP Package Service

Before configuring, consider the following:

`public_user` (if using Public Key Cryptography) or `conv_keymap_name` (if using Conventional Cryptography) must be present for PGP Package service to perform encryption.

`secret_keymap_name` must be present for PGP Package service to perform signing.

To perform encryption and signing, a combination of both the previous statements applies.

If `public_user` and `conv_keymap_name` appear in the same business process, public key encryption will take precedence.

To configure the PGP Package service, specify settings specify the settings for the fields in the GPM. These fields are described in the following table:

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files used for encryption and signing will be read from or written to. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line Adapter 2.
profile_name	Name of PGP profile from the PGP Profile Manager. Required.
compress	Compression to be done before encryption or signing. Valid value is On. Default is On. Required for encryption and signing.
public_user	User name or key ID in the public key ring. Required for encryption (public key cryptography).
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for signing (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for encryption (conventional cryptography).
conv_cipher	The symmetric cipher to use when performing a conventional encryption operation (that is, <code>conv_keymap_name</code> is used). Valid values are: IDEA, CAST5, 3DES, AES128, AES196, AES256, Twofish. Default is IDEA. Optional.

Field	Description
DocumentId	The document identifier referenced to the document to be processed specifically. The default document for processing is the primary document. Optional.
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
ascii_armor	Whether to encode the file with McAfee E-Business Server's base-64 encoding (ASCII-armored format). Valid values are On and Off. Default is On. Optional.
textmode	Whether the input data is ASCII text and should be converted to canonical new lines before encryption. Valid values are On and Off. Default is Off. Optional.
outputfilename	Output file name. For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc. For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of the document and is appended with the following: <ul style="list-style-type: none"> ◆ *.asc during normal encryption ◆ .exe during sda process ◆ .pga during pgparchive process Optional.
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.
clearsig	Generates a signed message that can be read without PGP. The recipient must still use PGP to verify the signature. Unencrypted PGP-signed messages have a signature certificate pre-pended in binary form. The signed message is compressed. Therefore, it is unreadable by humans even though it is not encrypted. Cannot be used with EncryptAndSign on the command line. If you enable clearsig, it is recommended you enable ascii_armor and textmode also. Valid values are On and Off. Default is Off. Optional.
info	How much information is returned. Valid values are: <ul style="list-style-type: none"> ◆ Quiet - Only displays error messages. Not applicable to PGP Command Line. If selected defaults to normal mode. ◆ Normal - Displays warnings and error messages. Default. ◆ Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). If selected with other versions, defaults to normal mode. ◆ Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line. If selected, defaults to normal mode. Optional.

Field	Description
sda	<p>Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified.</p> <p>Creates a self-decrypting executable file, which is conventionally encrypted using a passphrase. The resulting file can be decrypted by double-clicking it and entering the passphrase. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed.</p> <p>SDA files can be created with any platform that McAfee E-Business Server (version 8.1 or later) supports, but can be executed only on Windows platforms.</p> <p>To create sda files with PGP Command Line (version 9.5), set the target_platform parameter (described later in this table).</p> <p>The default file extension is .exe.</p> <p>Note: The sda file cannot exceed 4 GB after compression.</p> <p>Valid values are On and Off. Default is Off. Optional.</p>
pgparchive	<p>Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified.</p> <p>Creates a file that can be decrypted using the archive reader, which can be redistributed freely. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed.</p> <p>The default extension is .pga.</p> <p>Valid values are On and Off. Default is Off. Optional.</p>
discard_paths	<p>Applicable only with sda or pgparchive. Strips relative path information from the list of files in a sda or pgparchive. During the decryption of the archive, the files are placed in the current directory instead of in subdirectories of the current directory.</p> <p>Optional.</p>
target_platform	<p>Applicable only with PGP Command Line (version 9.5) and sda. Specifies the platform an sda file can be decrypted on. Valid values are:</p> <ul style="list-style-type: none"> ◆ win32 ◆ linux ◆ solaris ◆ aix ◆ hpux ◆ osx <p>Default is the current platform. Optional.</p>

Parameters Passed from Service to BP

The following table contains the parameters that are passed from the PGP Package service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are: <ul style="list-style-type: none"> ◆ ENCRYPT ◆ ENCRYPT_SIGN ◆ SIGN Required.
FileName (PGP/FileName)	Name of the file being processed. Required.
inputFileNamePkg (PGP/inputFileNamePkg)	Name of the file contained in the PGP package. Optional.
Document (PGP/Document)	The processed document is placed in Process Data – not as Primary Document. The attribute is the SCIOBJECTID, which enables a hyperlink for viewing the content of the processed document. Required.
DocumentId (PGP/DocumentId)	Document identifier of the document. Required.
Status (PGP/Status)	Process status. Valid values are Success and Error. Required.
ErrorCode (PGP/ErrorCode)	Value returned from executing PGP commands. Displayed when the Status is Error. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. Displayed when the Status is Error. Optional.

Business Process Example - Encrypt Operation (Public Key Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area. The profile is based on PGP107. In this example, you use the default Command Line2 adapter configuration, PGPCmdlineService, to execute the encrypt command. You want to use the working directory, remote name and port stated in the BPML. Therefore, these values override the pre-configured values in PGPCmdLineService. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption.

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
      </output message="Xout">
    </operation name="One">
  </sequence name="optional">
</process name="PGP_Encrypt ">
```



```

        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Encrypt Operation (Conventional Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area of process data. The profile is based on PGP107. In this example, you use the Command Line2 adapter configuration, MyCLA2, to execute the commands. The remote name, port, and working directory are pre-configured in the service configuration. The value of conv_keymap_name, Conv_abc_tp, which must be in the profile's conventional key map, is used for conventional encryption:

```

<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="conv_keymap_name">Conv_abc_tp</assign>
        <assign to="conv_cipher">CAST5</assign>
        <assign to="cmdline2svcname">MyCLA2</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - Encrypt and Sign Operation (Public Key Encryption)

The following business process uses the PGP Package service to encrypt and sign the primary document in the document area. For signing, you need to pass in the secret_keymap_name, which must be in the PGP107 profile's secret key map. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption. In this example, you choose not to compress the document before signing and encryption.

```

<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">off</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
        <assign to="secret_keymap_name">my_secret</assign>
</output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Encrypt and Sign Operation (Conventional Encryption)

The following business process uses PGP Package Service to encrypt and sign the Primary Document in the document area. For signing, the user needs to pass in the secret_keymap_name, which must be present in the PGP107 profile's Secret Key Map. The value of conv_keymap_name, Conv_abc_tp, which must be present in the Profile's Conventional Key Map, is used for conventional encryption. The user chooses not to compress the document before signing and encryption.

```

<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">off</assign>
        <assign to="workingDir">/localsvr/share/tmp</assign>
        <assign to="remoteName">nn.nnn.nn.nnn</assign>
        <assign to="remotePort">xxxxx</assign>
        <assign to="conv_keymap_name">Conv_abc_tp</assign>
        <assign to="conv_cipher">CAST5</assign>
        <assign to="secret_keymap_name">si_secret</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - Encrypt Operation (Public Key Encryption) Using a Specific Document ID

The following business process uses the PGP Package service to encrypt a document, with the document ID columbia:1774b9b:feaea8ae12:-6ea8 in the document area.

```

<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One"> PGPPackageService
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
        <assign to="public_user">0x2343</assign>
        <assign to="DocumentId">columbia:1774b9b:feaea8ae12:-6ea8</assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - Sign Operation

The following business process uses the PGP Package service to sign the primary document in the document area.

```

<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
        <assign to="secret_keymap_name">my_secret</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Business Process Example - OnFault Handling

The following business process shows the onFault handling for the PGP Package service.

```

<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="profile_name">PGP107</assign>
        <assign to="compress">on</assign>
        <assign to="workingDir">/localsvr/share/tmp</assign>
        <assign to="remoteName">nn.nnn.nn.nnn</assign>
        <assign to="remotePort">12345</assign>
        <assign to="secret_keymap_name">si_secret</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">

```

```

    <assign to="." from="*"></assign>
  </input>
</operation>
  <assign to="Status">The file is signed successfully</assign>
<onFault>
  <assign to="Status">General Error Occurred</assign>
</onFault>
<onFault code="[PGPErrorCode] Signature Check error">
  <assign to="Status">Incorrect signature</assign>
</onFault>
</sequence>
</process>

```

Advanced Status Messages

The following table contains exit codes from the McAfee E-Business Server and PGP Command Line Freeware. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.
160	Complete failure during a file wipe.
161	Partial fail, partial success during a file wipe (one file wiped, one not, for example).
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

The following table contains errors that result from the PGP Package service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId	The DocumentId value given in the BPML is incorrect.
Fail to get data from Primary Document.: There is no Primary Document	Primary Document is mandatory.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Profile Manager	The profile_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap	The secret_keymap_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap	The conv_keymap_name value given in the BPML is incorrect.

PGP Unpackage Service (Build 4300 - Build 4321)

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Unpackage service, in conjunction with the PGP Profile Manager, enables you to decrypt documents and verify their signatures.

The following table provides an overview of the PGP Unpackage service:

System name	PGP Unpackage service
Graphical Process Modeler (GPM) category	All Services
Description	This service is used to decrypt and verify the signature of a document based on the Open PGP standard, using a public key or conventional cryptography.
Business usage	Use this service to decrypt or verify the signature of the document in the document area.
Usage example	A business process is executed to decrypt or verify the signature of the document based on the PGP profile. See <i>PGP Profile Manager</i> .
Preconfigured?	Yes
Requires third-party files?	No

Platform availability	<p>All supported Application platforms, with the following restrictions:</p> <p>For NAI McAfee eBusiness Server 8.1:</p> <ul style="list-style-type: none"> ◆ IBM AIX 4.2 or later ◆ HP-UX 10.20 or later ◆ Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) ◆ SuSE Linux for IBM S/390 and IBM Zseries <p>For NAI McAfee eBusiness Server 8.5</p> <ul style="list-style-type: none"> ◆ Solaris 9 or later <p>For NAI McAfee eBusiness Server 8.5.1</p> <ul style="list-style-type: none"> ◆ Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) ◆ Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) ◆ Microsoft Windows Server 2003 ◆ Microsoft Windows XP Professional Version 2002 Service Pack 2 <p>For Massachusetts Institute of Technology (MIT) Command Line Freeware</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 ◆ UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later <p>For PGP Corporation PGP® Command Line 9.5</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 SP1 Microsoft Windows 2000 (SP 4) ◆ UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 ◆ Mac OS X 10.4 or greater
Related adapters	<p>The PGP Unpackage service works with the following services:</p> <ul style="list-style-type: none"> ◆ Command Line Adapter 2 ◆ PGP Package service

Application requirements	<p>Before using this service, install one of the following:</p> <ul style="list-style-type: none"> ◆ McAfee E-Business Server (version 8.1, 8.5, or 8.5.1) from Network Associates Technology, Inc. ◆ PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) ◆ PGP Command Line (version 9.5) from PGP Corporation <p>Note: Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.</p>
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	<p>Basic statuses are:</p> <ul style="list-style-type: none"> ◆ 0 - Success ◆ 1- Error <p>See <i>Advanced Status Messages</i> on page 1149 for a list of advanced statuses.</p> <p>Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrorCode].</p>
Restrictions	None
Persistence level	None
Testing considerations	<p>Create the profile in the PGP Profile Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.</p> <p>A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.</p> <p>To start the remote adapter implementation of the command line adapter:</p> <ol style="list-style-type: none"> 1 Locate the client jar (CLA2Client.jar) that contains all the necessary classes in the following directory: <ul style="list-style-type: none"> <code>install_DIR>/<client>/<cmdline2></code> 2 Move the client jar to the machine that has the PGP server installed. 3 Start the remote adapter implementation using the following command: <pre>java -jar CLA2Client.jar <port> [debug]</pre> <p>For example:</p> <pre>java -jar CLA2Client.jar 15699 debug</pre> <p>Note: The [debug] option is not required.</p>

Implement the PGP Unpackage Service

To implement the PGP Unpackage service, complete the following tasks:

1. Activate your license for the PGP Unpackage service.
2. Create a PGP profile, using the PGP Profile Manager.
3. Create a PGP Unpackage service configuration.
4. Configure the PGP Unpackage service.
5. Use the PGP Unpackage service in a business process.

Configure the PGP Unpackage Service

Before configuring the PGP Unpackage service, consider the following:

If the `secret_keymap_name` and `conv_keymap_name` parameters are not present, the PGP Unpackage service will verify the signature of the document only.

If one of the `keymap_name` parameters is present, it will use the information of the `keymap_name` to decrypt.

If there is a signature in the document, the verification of the signature will be done automatically.

To configure the PGP Unpackage service, specify the settings for the fields in the GPM. These fields are described in the subsequent table.

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files for decryption or verification will be read from or written to. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line 2 adapter. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line 2 adapter. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
profile_name	The name of PGP profile. Required.
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for decryption (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for decryption (conventional cryptography).
DocumentId	The document identifier for the document to be processed. The default document for processing is the primary document. Optional.

Field	Description
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
outputfilename	Output file name. For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc. For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of a document and is appended with the following: <ul style="list-style-type: none"> ◆ *.asc during normal encryption ◆ .exe during SDA process ◆ .pga during pgparchive process Optional.
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.
info	How much information is returned. Valid values are: <ul style="list-style-type: none"> ◆ Quiet - Only displays error messages. Not applicable to PGP Command Line (version 9.5). If selected, defaults to normal mode. ◆ Normal - Displays warnings and error messages. Default. ◆ Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). If selected with other versions, defaults to normal mode. ◆ Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line (version 9.5). If selected, defaults to normal mode. Optional.

The following table contains the parameters that are passed from the PGP Unpackage service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are DECRYPT and VERIFY. Required.
FileName (PGP/FileName)	The name of the file which is being processed. Required.
Document PGP/Document()	The processed document is placed in Process Data – not as Primary Document. The attribute is the SCIOBJECTID, which allows the user to click on it for viewing the content of the processed document. Required.

DocumentId (PGP/DocumentId)	The document identifier of the document. Required.
Status (PGP/Status)	The status shows if this process has completed successfully or failed. Valid values are Success and Error. Required.
ErrorCode PGP/ErrorCode()	This is the exit value returned from executing PGP commands. This will be shown when the Status is 'Error'. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. This will be shown when the Status is 'Error'. Optional.

Business Process Example - Decrypt Operation (Public Key Decryption)

The following business process uses the PGP Unpackage service to decrypt the primary document in the document area. The profile is based on PGP107. In this case, the default Command Line 2 adapter configuration, PGPCmdlineService, is used to execute the decrypt command. It uses the working directory, remote name and port stated in the business process. Therefore, these values will override any pre-configured values in PGPCmdlineService.

```
<process name="PGP_Decrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to=" secret_keymap_name"> my_secret </assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Verify Operation

The following business process uses the PGP Unpackage service to verify the primary document in the document area. The profile is based on PGP107. In this case, the Command Line 2 adapter configuration called MyCLA2 is used to execute the commands. The remote name, port and working directory have been pre-configured in the service configuration. Therefore, they are not required in the business process.

```
<process name="PGP_Verify">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

        <assign to="cmdline2svcname">MyCLA2</assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - OnFault Handling

The following business process shows onFault handling with the PGP Unpackage service.

```

<process name="PGP_Decrypt">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to=" secret_keymap_name"> si_secret </assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <assign to="Status">The file is decrypted successfully</assign>
  <onFault>
    <assign to="Status">General Error Occurred</assign>
  </onFault>
  <onFault code="[PGPErrorCode] Decryption error">
    <assign to="Status">Decryption error</assign>
  </onFault>
</sequence>
</process>

```

Advanced Status Messages

Exit Codes from E-Business Server and PGP Command Line Freeware

The following table contains exit codes from E-Business Server and PGP Command Line Freeware. The content of the description field will be displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found

Status	Description
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

Exit Codes from PGP Command Line - PGP Corporation

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the description field will be displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.

Status	Description
160	Complete failure during a file wipe.
161	Partial fail, partial success during a file wipe (one file wiped, one not, for example).
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

Errors During Validation

The following table contains errors that result from the PGP Unpackage service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId	The DocumentId value given in the bpml is incorrect.
Fail to get data from Primary Document	There is no Primary Document. Primary Document is mandatory.
You must enter one of these BPML Params: 'public_user' or 'secret_keymap_name' or 'conv_keymap_name'	Either one of the BPML Parameters must be present for PGP to encrypt, sign or encrypt and sign.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Profile Manager	The profile_name value given in the bpml is incorrect.
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap	The secret_keymap_name value given in the bpml is incorrect.

Status	Description
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap	The conv_keymap_name value given in the bpml is incorrect.

PGP Unpackage Service (Build 4322 or higher)

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Unpackage service, in conjunction with the PGP Profile Manager, enables you to decrypt documents and verify their signatures.

The following table provides an overview of the PGP Unpackage service:

System name	PGP Unpackage service
Graphical Process Modeler (GPM) category	All Services
Description	This service is used to decrypt and verify the signature of a document based on the Open PGP standard, using a public key or conventional cryptography.
Business usage	Use this service to decrypt or verify the signature of the document in the document area.
Usage example	A business process is executed to decrypt or verify the signature of the document based on the PGP profile. See <i>PGP Profile Manager</i> .
Preconfigured?	Yes
Requires third-party files?	No

Platform availability	<p>All supported Application platforms, with the following restrictions:</p> <p>For NAI McAfee eBusiness Server 8.1:</p> <ul style="list-style-type: none"> ◆ IBM AIX 4.2 or later ◆ HP-UX 10.20 or later ◆ Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) ◆ SuSE Linux for IBM S/390 and IBM Zseries <p>For NAI McAfee eBusiness Server 8.5</p> <ul style="list-style-type: none"> ◆ Solaris 9 or later <p>For NAI McAfee eBusiness Server 8.5.1</p> <ul style="list-style-type: none"> ◆ Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) ◆ Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) ◆ Microsoft Windows Server 2003 ◆ Microsoft Windows XP Professional Version 2002 Service Pack 2 <p>For Massachusetts Institute of Technology (MIT) Command Line Freeware</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 ◆ UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later <p>For PGP Corporation PGP® Command Line 9.5</p> <ul style="list-style-type: none"> ◆ Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 SP1 Microsoft Windows 2000 (SP 4) ◆ UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 ◆ Mac OS X 10.4 or greater
-----------------------	---

Related adapters	<p>The PGP Unpackage service works with the following services:</p> <ul style="list-style-type: none"> ◆ Command Line Adapter 2 ◆ PGP Package service
------------------	---

Application requirements	<p>Before using this service, install one of the following:</p> <ul style="list-style-type: none"> ◆ McAfee E-Business Server (version 8.1, 8.5, 8.5.1, or 8.6) from Network Associates Technology, Inc. ◆ PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) ◆ PGP Command Line (version 9.5) from PGP Corporation <p>Note: Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.</p>
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	<p>Basic statuses are:</p> <ul style="list-style-type: none"> ◆ 0 - Success ◆ 1- Error <p>See <i>Advanced Status Messages</i> on page 1159 for a list of advanced statuses.</p> <p>Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrCode].</p>
Restrictions	None
Persistence level	None
Testing considerations	<p>Create the profile in the PGP Profile Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.</p> <p>A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.</p> <p>To start the remote adapter implementation of the command line adapter:</p> <ol style="list-style-type: none"> 1 Locate the client jar (CLA2Client.jar) that contains all the necessary classes in the following directory: <ul style="list-style-type: none"> <code>install_DIR>/<client>/<cmdline2></code> 2 Move the client jar to the machine that has the PGP server installed. 3 Start the remote adapter implementation using the following command: <pre>java -jar CLA2Client.jar <port> [debug]</pre> <p>For example:</p> <pre>java -jar CLA2Client.jar 15699 debug</pre> <p>Note: The [debug] option is not required.</p>

Implement the PGP Unpackage Service

To implement the PGP Unpackage service, complete the following tasks:

1. Activate your license for the PGP Unpackage service.
2. Create a PGP profile, using the PGP Profile Manager.
3. Create a PGP Unpackage service configuration.
4. Configure the PGP Unpackage service.
5. Use the PGP Unpackage service in a business process.

Configure the PGP Unpackage Service

Before configuring the PGP Unpackage service, consider the following:

If the `secret_keymap_name` and `conv_keymap_name` parameters are not present, the PGP Unpackage service will verify the signature of the document only.

If one of the `keymap_name` parameters is present, it will use the information of the `keymap_name` to decrypt.

If there is a signature in the document, the verification of the signature will be done automatically.

To configure the PGP Unpackage service, specify the settings for the fields in the GPM. These fields are described in the subsequent table.

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files for decryption or verification will be read from or written to. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line 2 adapter. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the <code>cmdline2svcname</code> field is defined in the Command Line 2 adapter. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
profile_name	The name of PGP profile. Required.
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for decryption (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for decryption (conventional cryptography).
DocumentId	The document identifier for the document to be processed. The default document for processing is the primary document. Optional.

Field	Description
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
outputfilename	Output file name. For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc. For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of a document and is appended with the following: <ul style="list-style-type: none"> ◆ *.asc during normal encryption ◆ .exe during SDA process ◆ .pga during pgparchive process Optional.
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.
info	How much information is returned. Valid values are: <ul style="list-style-type: none"> ◆ Quiet - Only displays error messages. Not applicable to PGP Command Line (version 9.5). If selected, defaults to normal mode. ◆ Normal - Displays warnings and error messages. Default. ◆ Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). If selected with other versions, defaults to normal mode. ◆ Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line (version 9.5). If selected, defaults to normal mode. Optional.

The following table contains the parameters that are passed from the PGP Unpackage service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are DECRYPT and VERIFY. Required.
FileName (PGP/FileName)	The name of the file which is being processed. Required.
Document PGP/Document()	The processed document is placed in Process Data – not as Primary Document. The attribute is the SCIObjctID, which allows the user to click on it for viewing the content of the processed document. Required.

DocumentId (PGP/DocumentId)	The document identifier of the document. Required.
Status (PGP/Status)	The status shows if this process has completed successfully or failed. Valid values are Success and Error. Required.
ErrorCode PGP/ErrorCode()	This is the exit value returned from executing PGP commands. This will be shown when the Status is 'Error'. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. This will be shown when the Status is 'Error'. Optional.

Business Process Example - Decrypt Operation (Public Key Decryption)

The following business process uses the PGP Unpackage service to decrypt the primary document in the document area. The profile is based on PGP107. In this case, the default Command Line 2 adapter configuration, PGPCmdlineService, is used to execute the decrypt command. It uses the working directory, remote name and port stated in the business process. Therefore, these values will override any pre-configured values in PGPCmdlineService.

```
<process name="PGP_Decrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to=" secret_keymap_name"> my_secret </assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Verify Operation

The following business process uses the PGP Unpackage service to verify the primary document in the document area. The profile is based on PGP107. In this case, the Command Line 2 adapter configuration called MyCLA2 is used to execute the commands. The remote name, port and working directory have been pre-configured in the service configuration. Therefore, they are not required in the business process.

```
<process name="PGP_Verify">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

        <assign to="cmdline2svcname">MyCLA2</assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>
</sequence>
</process>

```

Business Process Example - OnFault Handling

The following business process shows onFault handling with the PGP Unpackage service.

```

<process name="PGP_Decrypt">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPUnPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="profile_name">PGP107</assign>
        <assign to="secret_keymap_name">si_secret</assign>
        <assign to="workingDir">/server1/tmp</assign>
        <assign to="remoteName">00.000.00.000</assign>
        <assign to="remotePort">12345</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <assign to="Status">The file is decrypted successfully</assign>
  <onFault>
    <assign to="Status">General Error Occurred</assign>
  </onFault>
  <onFault code="[PGPErrorCode] Decryption error">
    <assign to="Status">Decryption error</assign>
  </onFault>
</sequence>
</process>

```

Advanced Status Messages

Exit Codes from E-Business Server and PGP Command Line Freeware

The following table contains exit codes from E-Business Server and PGP Command Line Freeware. The content of the description field will be displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found

Status	Description
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

Exit Codes from PGP Command Line - PGP Corporation

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the description field will be displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.

Status	Description
160	Complete failure during a file wipe.
161	Partial fail, partial success during a file wipe (one file wiped, one not, for example).
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

Errors During Validation

The following table contains errors that result from the PGP Unpackage service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId	The DocumentId value given in the bpml is incorrect.
Fail to get data from Primary Document	There is no Primary Document. Primary Document is mandatory.
You must enter one of these BPML Params: 'public_user' or 'secret_keymap_name' or 'conv_keymap_name'	Either one of the BPML Parameters must be present for PGP to encrypt, sign or encrypt and sign.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Profile Manager	The profile_name value given in the bpml is incorrect.
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap	The secret_keymap_name value given in the bpml is incorrect.

Status	Description
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap	The conv_keymap_name value given in the bpml is incorrect.

Pre 3.1 Self Registration Service

The following table provides a high-level overview of the Pre 3.1 Self Registration Service:

System name	Pre 3.1 Self Registration Service
Graphical Process Modeler (GPM) categories	SyncMode > Self Registration
Description	This service is used to allow the caller to update a user's password.
Business usage	This service allows you to update a user's password in the database.
Usage example	A Gentran Integration Suite business process is executed that requires updating a user's password in database. The Pre 3.1 Self Registration Service is used to update the specified user's password in the system.
Preconfigured?	The Pre 3.1 Self Registration Service must be installed and deployed before it can be invoked.
Requires third party files?	No
Platform availability	This adapter is available on these platforms: <ul style="list-style-type: none">◆ Microsoft Windows 2000◆ Sun Solaris◆ HP-UX◆ IBM-AIX◆ United Linux◆ RedHat AS◆ iSeries (OS/400)◆ zSeries (z/OS)
Related services	Self Registration Service
Application requirements	The user must exist in the system and the old Password field must match the user's current password in the system.
Initiates business processes?	None
Invocation	Runs as part of a business process.
Business process context considerations	The user must exist in the system and the old Password field must match the user's current password in the system.

Returned status values	<ul style="list-style-type: none"> ◆ 0 - Success ◆ 1 - Error
Restrictions	None
Persistence level	Default
Testing considerations	Debug information for this service can be found in the system log files.

Configuration Parameters

Parameter (Display Name)	Description
Name	The name that the service will have in Gentran Integration Suite.
Description	Description of the service.
Select a Group	<p>A button that allows selection or creation of the group that this service will be part of.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ None ◆ CreateNewGroup ◆ SelectGroup

Parameters Passed from Business Process to Service

Parameter (Display Name)	Description
username	<p>The username of the password you want to update.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Any valid existing user name
oldPassword	<p>The old username password.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Only valid current password.
newPassword	<p>The new password you want to update for the username.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Any valid new password
retypedPassword	<p>The new password you want to update for the username.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Any valid new password

Example Business Process

```
<process name="testuser">
  <sequence name="testuser">
    <operation name="testuser">
      <participant name="SelfRegistration"/>
      <output message="Xout">
        <assign to="username">testuser</assign>
        <assign to="oldPassword">P@$$Word</assign>
        <assign to="newPassword">NewP@$$Word</assign>
        <assign to="retypedPassword">NewP@$$Word</assign>
      </output>
      <input message="Xin">
      </input>
    </operation>
  </sequence>
</process>
```

Purge Business Process Linkage Service

Caution: This is an internal service that should not be used externally for steps in creating business processes because it is subject to change without notice, and may cause unpredictable results and loss of data. This section is intended for information purposes only.

The following table provides an overview of the Purge Business Process Linkage service:

System name	BPLinkagePurgeServiceType
Graphical Process Modeler (GPM) categories	All Services, System
Description	System service used when a business process needs to remove business process instance hierarchical data from WORKFLOW_LINKAGE table.
Business usage	This service checks all hierarchical data in WORKFLOW_LINKAGE table, if all business processes of a hierarchical have been archived or purged, then it removes this hierarchical linkage records from the table. This service can be scheduled during configuration.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Archive Business Process service, Index Business Process service.
Application requirements	No
Initiates business processes?	No
Invocation	Generally, this service is invoked by the Scheduler. By default, the service runs every day at 2:00 am.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	Cannot have multiple configurations. This is an internal service, and should not be used in a business process.

Purge Service

The Purge service looks for eligible records to delete from the Live system databases. The following table provides an overview of the Purge service:

System name	PurgeServiceType
Graphical Process Modeler (GPM) categories	All Services, System
Description	Looks for eligible records to delete from the Live system database.
Business usage	<p>Deletes rows by controlling the number of rows to use from ARCHIVE_INFO when purging tables. The Purge Service continues to loop until the purge is complete. This gives you control over how much data can be deleted for each database transaction and also makes sure that all data for a given workflowid is deleted from a table in one transaction.</p> <p>The Purge service physically removes records that have been flagged as eligible for deletion from the Live system databases. Eligibility is determined by the Index Business Process service, which scans the Live systems and flags records that have reached their purge eligibility date and time.</p>
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Index Business Process service, Archive service, Purge BP Linkage service
Application requirements	No
Initiates business processes?	No
Invocation	Generally, this service is invoked by the Scheduler. The default schedule is to run every ten minutes, but the frequency can be changed.
Business process context considerations	No
Restrictions	This is an internal service, and should not be used in a business process.

How the Scheduled Purge Works

The purge process takes a specified number of rows from a table for purging, which enables more control over how much data is purged from a table in one transaction. The system takes the first 1000 business processes and deletes all of the data in a specific table for that business process ID. A “loop” is the complete purge of all the tables for those 1000 business processes.

If there is more data than can be deleted in one transaction, due to the row limitation, the purge process continues to loop until all eligible data has been purged. For most users, this setting will not need to be changed; however, if you have very high data loads, you may want to consult Sterling Commerce Customer Support for information about customizing this process.

There is no configuration necessary for the Purge service; the service is installed with Application, and scheduled to run every ten minutes. Unless you need to change the schedule, there is no setup required.

Purge All Business Process

The Purge service includes the ability to run a Purge All command. Note that this functionality should only be used when requested by Sterling Commerce Customer Support. It enables you to purge all eligible records, regardless of date, on demand. The service uses a new BP, PurgeAll.bp, which contains two flags, Purge (set to ALL) and Max Loops (set by default to 100). You can change the Max Loops value. The Purge All and scheduled Purge functions cannot run at the same time, so if the regularly scheduled Purge service is running, the Purge All activity will not start until the scheduled purge is done. If the Purge All activity is running, the scheduled purge will not run until the Purge All is done. (In these cases, you do not need to take any action – this is just for your information. You may want to disable the purge schedule before starting a Purge All, but it is not required.)

Caution: The Purge All business process should not be used for ordinary production purposes. It is only for use, generally on the advice of Sterling Customer Support, to immediately remove data from the live system, regardless of its expiration date. This may be advisable, for example, if the Scheduled Purge business process has encountered some failure causing a back up of purge-eligible data. There is an additional flag (MAX_LOOPS) available that will help limit the number of loops made by the Purge All business process, thereby helping to control how much data the system will handle in a single execution. If a large amount of data had accumulated, this limit will help the system continue with other processing. To contact Customer Support, go to www.sterlingcommerce.com and follow the links to Customer Support.

The following example shows the PurgeAll business process as delivered with Application. Application replaces the necessary entities with the correct values for your system when the business process is invoked.

```
<process name="PurgeAll">
  <rule name="obtainLock">
    <condition>controlLock ='true' </condition>
  </rule>
  <sequence>
    <assign to='controlLock'>>false</assign>
  <operation name="SetLock">
    <participant name="SystemLockService"/>
    <output message="Xout">
      <assign to="LOCK_KEY">PURGE</assign>
      <assign to="DURATION">86400000</assign>
    <assign to="CLEAR_ON_START_UP">>true</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>
  <assign to='controlLock'>>true</assign>
  <operation name="Purge Service">
    <participant name="PurgeService"/>
    <output message="Xout">
      <assign to="PURGE_MODE">ALL</assign>
      <assign to="MAX_LOOPS">100</assign>
      <assign to="." from="*"></assign>
    </output>
  </operation>
</process>
```



```

        </output>
        <input message="Xin">
            <assign to="." from="*"></assign>
        </input>
    </operation>

<operation name="unLock">
    <participant name="SystemLockService"/>
    <output message="Xout">
        <assign to="ACTION">unlock</assign>
        <assign to="LOCK_KEY">PURGE</assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>

<onFault>
    <sequence name="LockFailed">
        <choice>
            <select>
                <case ref="obtainLock" activity="proceedWithLocking"/>
                <case ref="obtainLock" negative="true" activity="stopWithoutLocking"/>
            </select>
            <sequence name="proceedWithLocking">
                <operation>
                    <participant name="SystemLockService"/>
                    <output message="Xout">
                        <assign to="ACTION">unlock</assign>
                        <assign to="LOCK_KEY">PURGE</assign>
                        <assign to="." from="*"></assign>
                    </output>
                    <input message="Xin">
                        <assign to="." from="*"></assign>
                    </input>
                </operation>
            </sequence>
            <sequence name="stopWithoutLocking">
                <assign to="Document/Msg" append="true">Failed to obtain a lock.!!</assign>
            </sequence>
        </choice>
        <assign to="Document/Status" append="true">Failed!</assign>
        <assign to="Document/Msg" append="true">Purging failed!</assign>
        <assign to="Document/CurrentHost" append="true">&HOST_NAME;</assign>
        <assign to="Document/CurrentPort" append="true">&LIST_PORT;</assign>
        <assign to="Document/DetailMsg" from="/ProcessData/StatusRpt/text()"
append="true"></assign>
        <operation name="SMTP Send">
            <participant name="SMTP_SEND_ADAPTER"/>
            <output message="SMTP_SEND_ADAPTERInputMessage">
                <assign to="xport-smtp-mailfrom">&SI_ADMIN_MAIL_ADDR;</assign>
                <assign to="xport-smtp-mailhost">&SI_ADMIN_SMTP_HOST;</assign>
                <assign to="xport-smtp-mailport">25</assign>
                #ifdef WOODSTOCK_DIR
                <assign to="xport-smtp-mailto">alert_test@stercomm.com</assign>
            </output>
        </operation>
    </sequence>
</onFault>

```

```

#:else
    <assign to="xport-smtp-mailto">&SI_ADMIN_MAIL_ADDR;</assign>
#:endif
    <assign to="xport-smtp-mailsubject">Automated Event Notification - Purging
All Failed</assign>
    <assign to="PrimaryDocument" from="DOMToDoc(Document)/@*"></assign>
    <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
    <assign to="." from="*"></assign>
    </input>
    </operation>
</sequence>
</onFault>
</sequence>
</process>

```

Purge Business Process Chunk

The purge service deletes the rows in a chunk instead of a single delete statement to prevent lock timeouts, excessive rollback, and so on. The chunk parameter in purge service is defined by the number of business processes (Workflow IDs) the process tries to delete in each iteration. The process will go through each table and deletes all rows associated with that group of business processes. The default value is 2000. The purge service finds all business processes that are currently marked as eligible for purging and whose date is less than the system time. It then deletes them in groups using the chunk parameter until they have all been deleted.

The following example shows the purge service defined to delete rows in groups using the chunk parameter:

```

<operation name="Purge Service">
    <participant name="PurgeService"/>
    <output message="Xout">
        <!-- Added for infinite loop chunk purge -->
        <assign to="CHUNK">50</assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>

```

Release Service

The Release service is a system service used to remove elements from the process data of a business process. The following table provides an overview of the Release service:

System name	ReleaseService
Graphical Process Modeler (GPM) categories	All Services, System
Description	The Release service is a system service used to remove elements from the process data of a business process.
Business usage	None – System service
Usage example	Use the Release service to remove elements that are no longer needed by a business process. Some of the benefits for using the Release service include: <ul style="list-style-type: none">◆ Freeing up the storage space used by elements being removed◆ Removing “clutter” from process data, making it easier to understand processing and debug problems◆ Reducing the size of process data, which can speed the execution of XPath functions that work on process data
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms.
Related services	Assign activity – Used to put data into process data for a business process.
Application requirements	None
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – If release is successful◆ Failure – If release is unsuccessful
Testing considerations	None (system service)

How the Release Service Works

The Release service is a system service used to remove information from process data in business processes. See *Usage Examples* on page 1172 for examples using the Release service.

Implementing the Release Service

Because the Release service is preconfigured, you only need to include the service in a business process and define the parameters for the Release service to operate on.

Release Service Parameters

The following table describes the Release service parameter:

Parameter	Description
TARGET	<p>The element(s) to remove from process data.</p> <p>Notes:</p> <ul style="list-style-type: none">◆ You must specify the TARGET parameter in all uppercase characters.◆ You can use either the Advanced Editor in the GPM or BPML to specify this parameter.◆ You can use the following formats to assign an element to the TARGET parameter. <pre><assign to="TARGET">target_element_name</assign></pre> <p>where <i>target_element_name</i> is the name of the XML element you want removed from process data. The Use XPATH? option in the GPM is not selected when using constants to assign the TARGET parameter.</p> <p>or</p> <pre><assign to="TARGET" from=" 'xpath_statement' " /></pre> <p>Notice that the <i>xpath_statement</i> is enclosed in both double quotes and single quotes. The Use XPATH? option in the GPM must be selected when using XPath statements to assign the TARGET parameter.</p> <p>See <i>Usage Examples</i> on page 1172 for examples.</p>

Usage Examples

The following are examples using the Release service using both the GPM and BPML.

Removing Temporary Variables from Process Data

The following example using BPML illustrates a business process that uses the Release service to remove temporary variables from process data.

```

<process name="ExampleReleaseTemporaryVariables">
  <sequence name="Sequence Start">
    <assign to="TempVariables/Variable1" from="'1'" append="true"></assign>
    <assign to="TempVariables/Variable2" from="'2'"></assign>
    <assign to="TempVariables/Variable3" from="'3'"></assign>
  </sequence>
  <operation name="release">
    <participant name="ReleaseService" />
    <output message="outmsg">
      <assign to="TARGET">TempVariables</assign>
    </output>
    <input message="inmsg" />
  </operation>
</process>

```

Each **assign** statement assigns a temporary variable in process data under the element TempVariables.

TARGET specifies TempVariables as the element to remove from process data.

After the assigns are processed, the process data looks like the following:

```

<ProcessData>
  <TempVariables>
    <Variable1>1</Variable1>
    <Variable2>2</Variable2>
    <Variable3>3</Variable3>
  </TempVariables>
</ProcessData>

```

After the Release service is processed, the process data looks like the following, removing the temporary variables:

```
<ProcessData/>
```

The following example using the GPM illustrates the same business process.

ExampleReleaseTemporaryVariables.hp

Service Editor-release

Name: release

Config: ReleaseService

Message To Service	Message From Service
Output Msg	Messages Only
Message Name	outmsg
Name	

Advanced Editor: release

Name	Value	Use XPath?
TARGET	TempVariables	<input type="checkbox"/>

Specify the TARGET parameter and its corresponding value.

OK Cancel

Click **Advanced** to open the Advanced Editor.

Advanced

Removing a Primary Document Reference from Process Data

The following example illustrates a business process that uses the Release service to remove the primary document reference from process data.

The following document is input to the business process and becomes the primary document:

```
<Order Id="200">
  <Customer Id="222">
    <Name>John Doe</Name>
    <Address>
      <Street>888 Cherry Street</Street>
      <City>Denver</City>
      <State>CO</State>
      <Zip>80265</Zip>
    </Address>
    <Phone>303-629-5555</Phone>
  </Customer>
  <Account Id="800">
    <Customer_Id>555</Customer_Id>
    <Name>XXX Account</Name>
    <Type>Personal</Type>
    <Contact>
      <Name>Jane Doe</Name>
      <Phone>720-555-1234</Phone>
    </Contact>
  </Account>
  <Order_Item Id="400">
    <Order_Id>100</Order_Id>
    <Name>XXX</Name>
    <Quantity>1</Quantity>
    <Price> 39.99</Price>
  </Order_Item>
</Order>
```

The following example illustrates the business process using BPML.

```
<process name="ExampleReleasePrimaryDocument">
  <sequence name="main">
    <operation name="release">
      <participant name="ReleaseService"/>
      <output message="outmsg">
        <assign to="TARGET"/></ProcessData/PrimaryDocument</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg"/>
    </operation>
  </sequence>
</process>
```

TARGET specifies /ProcessData/
PrimaryDocument as the element to
remove from process data.

After the business process runs, the process data looks like the following, with a reference to the primary document:

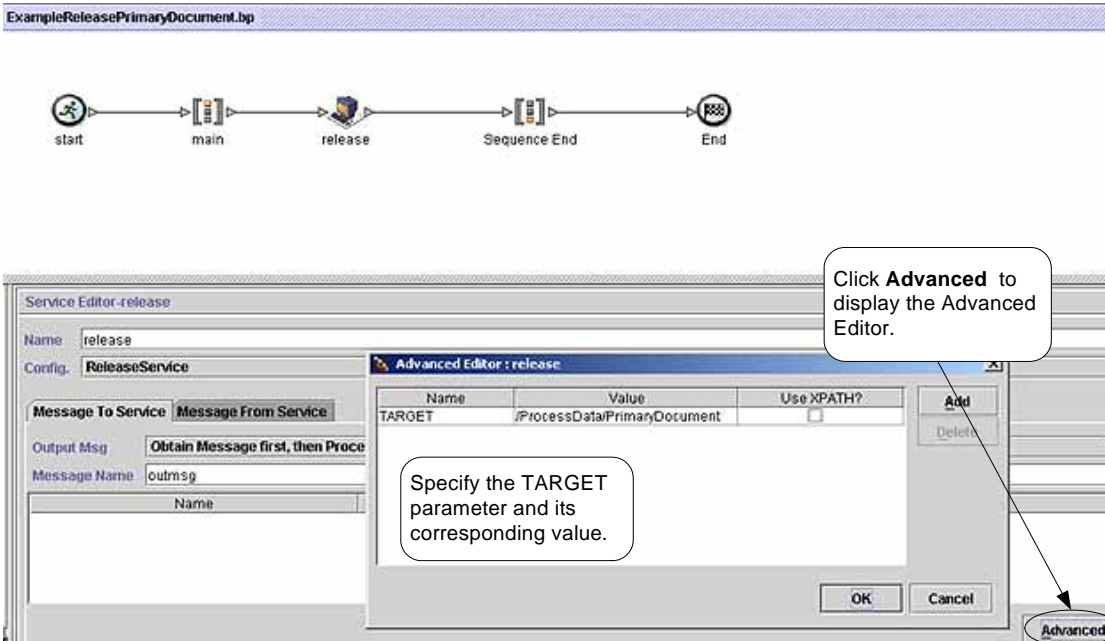
```
<ProcessData>
  <PrimaryDocument SCIObjectID="belushi:712b3a:fb0cd0a610:3255" />
</ProcessData>
```

Note: If you click on the SCIOBJECTID, you see the actual primary document.

After the Release service is processed, the process data looks like the following, removing the reference to the primary document:

```
<ProcessData/>
```

The following example illustrates the business process using the GPM:



Removing a Temporary Document from Process Data

The following example illustrates a business process that does the following:

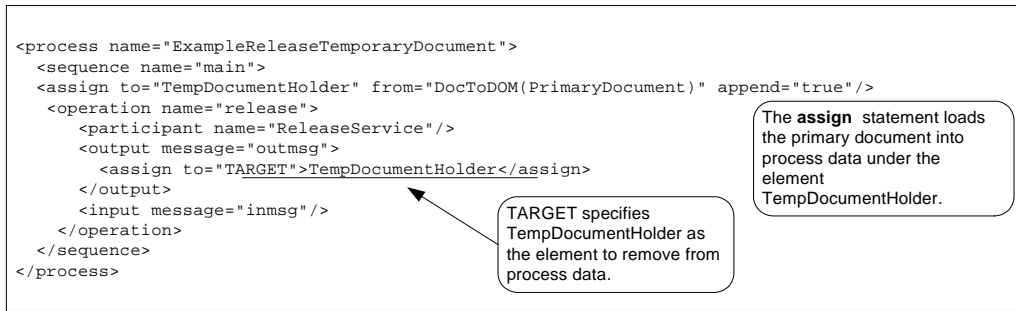
- Loads a primary document into process data but maintains a reference to the original primary document.

- Removes the document from process data, keeping the original primary document reference in process data.

You might consider this approach when you want the business process to manipulate the data contained in the primary document and yet maintain the original primary document.

The document used in the previous example is input to the business process and becomes the primary document. See *Removing a Primary Document Reference from Process Data* on page 1174 for the input document.

The following example illustrates the business process using BPML:



After the business process runs and the assign statement is processed, the process data looks like the following:

```

<ProcessData>
  <PrimaryDocument SCIObjectID="server1:712b3a:fb0cd0a610:35bd" />
  <TempDocumentHolder>
    <Order Id="200">
      <Customer Id="222">
        <Name>John Doe</Name>
        <Address>
          <Street>888 Cherry Street</Street>
          <City>Denver</City>
          <State>CO</State>
          <Zip>80265</Zip>
        </Address>
        <Phone>303-629-5555</Phone>
      </Customer>
      <Account Id="800">
        <Customer_Id>555</Customer_Id>
        <Name>XXX Account</Name>
        <Type>Personal</Type>
        <Contact>
          <Name>Jane Doe</Name>
          <Phone>720-555-1234</Phone>
        </Contact>
      </Account>
      <Order_Item Id="400">
        <Order_Id>100</Order_Id>
        <Name>XXX</Name>
        <Quantity>1</Quantity>
        <Price> 39.99</Price>
      </Order_Item>
    </Order>
  </TempDocumentHolder>
</ProcessData>

```

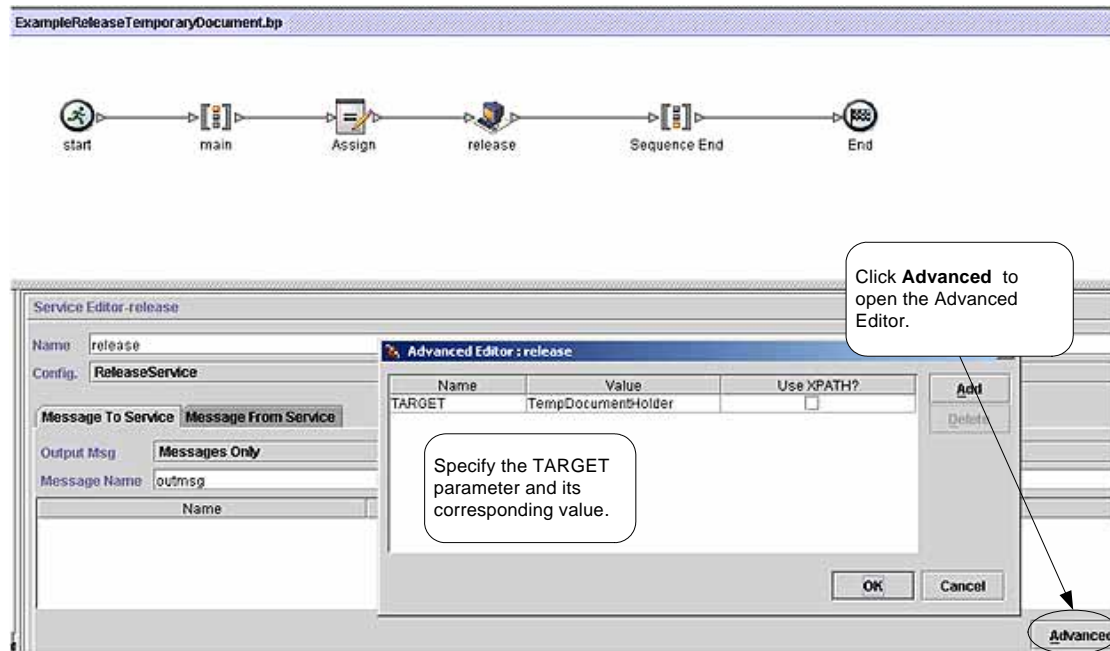
After the Release service is processed, the process data looks like the following, with references to the original primary document and the temporary document removed:

```

<ProcessData>
  <PrimaryDocument SCIObjectID="server1:712b3a:fb0cd0a610:35bd" />
</ProcessData>

```


The following example illustrates the business process using the GPM:



Best Practices

This section lists some best practices to follow when using the Release service to increase the efficiency of your business process.

Organize the creation of elements in process data in a way that simplifies their removal. For example, use a wrapper element to enclose the elements you add to process data.

Use one Release service to release multiple elements in process data by setting the TARGET parameter using a function like name(). The following example removes the three named elements from process data:

```
<assign to="TARGET">/ProcessData/TempDocumentHolder/Order/*[name()= 'Customer' or name()= 'Account' or name()= ' Order_Item']</assign>
```

Remote Method Invocation (RMI) Adapter

Note: In the Application interface, log files, BPML, service configuration, and in the class names of the code you write to use the adapter, this adapter is called the RMI IIOP adapter, or RMIIOPAdapter. This name is a reflection of its former use with Internet Inter-Orb Protocol (IIOP). IIOP is no longer supported.

The following table provides an overview of the RMI adapter:

System name	None
Graphical Process Modeler (GPM) category	None
Description	<p>Receives requests from remote RMI clients to start business processes. You can start a business process within Application from your remote system and get the results immediately or wait for the results. The RMI adapter also enables you to check the status of a business process.</p> <p>The RMI client can use the RMI adapter over the Java Remote Method Protocol (JRMP).</p>
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	<p>The remote client can use RMI over JRMP to access this adapter. It must have the appropriate application server jar file in its classpath (for example, weblogic.jar for WebLogic). It must also have in its classpath the jar file containing the RMI stub class.</p> <p>Typically, the system properties <code>java.naming.factory.initial</code> and <code>java.naming.provider.url</code> are configured to select JRMP. See your application server documentation for more information.</p>
Initiates business processes?	<p>Provides the remote RMI Client with the following two options for starting a business process:</p> <ul style="list-style-type: none">◆ Start a business process and wait for its completion◆ Start a business process and return immediately with its business process ID
Invocation	<p>This adapter starts business processes but it does not run by business processes. When starting a business process through this adapter, remote RMI clients may pass parameters to that business process as key-value pairs.</p>
Business process context considerations	No

Returned status values	The RMI adapter can return one of the following five statuses to the RMI Client: <ul style="list-style-type: none">◆ Success◆ Failure◆ Interrupted◆ In-Progress◆ Unknown
Restrictions	The RMI adapter can only start a business process. It can send a document or key/value pairs as inputs to the business process. Business process completion status and business process resultant documents may be retrieved by the RMI client using the RMI adapter. This adapter should never be used inside an Application business process.

Requirements

The RMI adapter has the following requirements and restrictions:

For security reasons, the remote client should be in the same secured area as Application. There is no automatic secure channel associated with RMI.

If Application and the remote client are not in the same secured area, configure JRMP to run over SSL to provide a secure channel.

The remote client can use RMI over JRMP to access the RMI adapter.

The remote client must have the appropriate application server .jar file in its classpath, for example, weblogic.jar for WebLogic.

The remote client must also have in its classpath the jar file containing the RMI stub class. Typically, the system properties *java.naming.factory.initial* and *java.naming.provider.url* are configured to select JRMP. See your application server documentation for more details. For more information about .jar files, see *Application Server .jar Files* on page 1184.

WebLogic and JBoss support only RMI over JRMP.

How the RMI Adapter Works

The RMI adapter works with synchronous or asynchronous requests. This section explains how the RMI adapter functions for each type of request.

Asynchronous Requests

The following procedure illustrates how the RMI adapter handles asynchronous requests:

1. The RMI adapter receives a request to start a business process from an RMI client, for example, `runBPDocParmNoWait()`. This request contains any business process parameters and documents that the business process might need.
2. The RMI adapter starts a user-defined business process, and passes any input documents and business process parameters that were received with the request.

3. The RMI adapter ends the business process and returns control to the RMI client. The RMI client can later poll the RMI adapter (using the business process ID) to check the completion of the business process, and to retrieve a resulting document, if applicable.

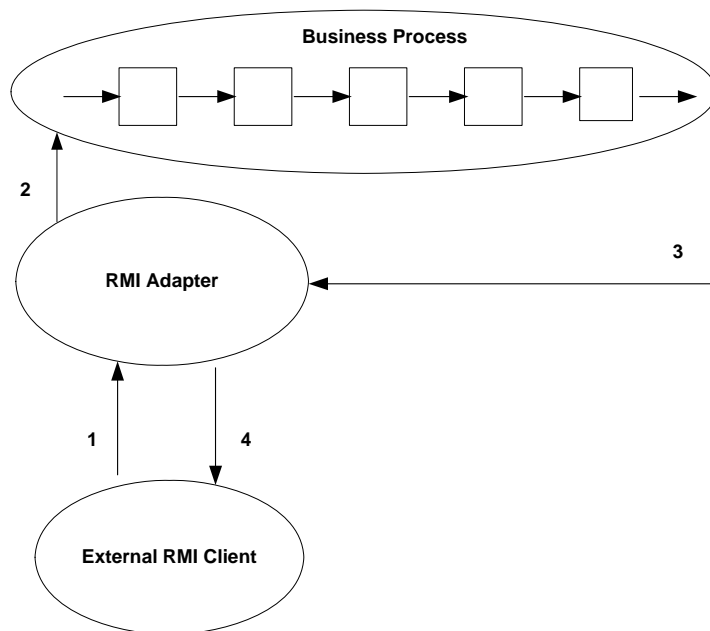
Synchronous Requests

The following procedure illustrates how the RMI adapter handles synchronous requests:

1. The RMI adapter receives a request to start a business process from an RMI client, for example, `runBPDocParmWait()`. This request contains any business process parameters and documents that the business process might need.
2. The RMI adapter starts a user-defined business process, and passes any input documents or business process parameters that were received with the request.
3. The business process completes and returns to the RMI adapter. The RMI adapter now has the final status of the business process and any associated documents.
4. The RMI adapter returns to the RMI client an object (`InvokedBusinessProcessDetails`) containing the business process ID, business process status, and any resulting documents.

Not all business processes have documents associated with them. If a business process has multiple documents, only the primary document is returned.

The following figure shows the RMI adapter as it communicates with the external RMI client during a synchronous request:



Implementing the RMI Adapter

To implement the RMI adapter, complete the following tasks:

1. Create an RMI adapter configuration. For information, see *Managing Services and Adapters*.

2. Configuring the RMI adapter. For information, see *Configuring the RMI Adapter* on page 1181.
3. Specify RMI method parameters. For information, see *Starting a Business Process* on page 1181.
4. Set system properties to make connections to the RMI adapter. For information, see *Making Connections to the RMI Adapter* on page 1181.
5. Use one of the provided RMI methods to connect to the RMI adapter and to complete a synchronous or asynchronous request. For information, see *Business Process Requests* on page 1181.
6. Use the RMI adapter in a business process.

Configuring the RMI Adapter

Because the RMI adapter is inbound and is not started by a business process, you need to configure the adapter one time only. This one configuration serves multiple clients. There are no configuration parameters other than providing a name and description for the RMI adapter.

Caution: Creating more than one configuration of the RMI adapter can cause unpredictable results.

Starting a Business Process

The RMI adapter needs the following information to start a business process. You can supply this information as parameters to an RMI method.

Business process name – Unique name of an existing business process. Required.

Business process document – Document to be passed to the business process. Optional.

Name-value pairs – Name-value pairs placed at the top level of the process data for the business process being started. Optional.

Time out value – Length of time to wait for a business process to complete. Optional.

Security credentials – Data needed for authentication. Required.

Making Connections to the RMI Adapter

On the client Java Virtual Machine (JVM), set the following system properties for WebLogic RMI over JRMP:

```
java.naming.factory.initial=org.jnp.interfaces.NamingContextFactory
```

```
java.naming.provider.url=jnp://100.100.100.100:1000
```

```
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
```

The following code retrieves the interface for starting the RMI adapter methods:

```
//Get an Initial Naming Context
Context initialNamingContext = new InitialContext();
//Find the RMI Adapter in the JNDI tree
Object obj=initialNamingContext.lookup("IIOPBusinessProcessProxy");
//Narrow the retrieved object to the interface for Invoking Business Processes
IBusinessProcessProxy proxy=(IBusinessProcessProxy)
    PortableRemoteObject.narrow(obj, IBusinessProcessProxy.class);
```

Business Process Requests

The RMI adapter provides several RMI methods that the remote RMI client can use to start business processes and pass documents and business process parameters to those business processes.

The following code examples show how to connect to the RMI adapter and how to start a synchronous or asynchronous request.

Invoke Business Process and Wait – The `runBPDocWait()` method call (synchronous request) runs on the server (or adapter). The call will not complete until the business process completes.

```
RMIIOPAdapterDocument doc=new RMIIOPAdapterDocument();
//Set up document to pass to business process
doc.setName(document name);
doc.setSubject(document subject);
doc.setContentType(document content type);
doc.setContentSubType(document content subtype);
doc.setBody("documentbodyishere".getBytes());

//Invoke the business process, giving it an initial document and wait for
completion.
//It will wait for the default timeout period. See BusinessProcessProxy for
the default timeout period.
InvokedBusinessProcessDetails bpDetails=null;
try {
    bpDetails=proxy.runBPDocWait(NameOfTheBPToInvoke, doc,
securityCredentials);
}
catch(Exception e) {
    //If a Timeout occurs then a BusinessProcessException is returned with
TIMEOUT_EXCEEDED.
    System.out.println("Security, Remote, or BusinessProcess exception has
occurred");
    System.out.exit(-1);
}
//Get the returned document
String returnedDocument = null;
if(bpDetails.getBody()!=null) {
    returnedDocument=new String(bpDetails.getBody());
}
```

Invoke Business Process and Return Immediately – The method call, `runBPNoWait` (asynchronous request), starts the business process and immediately returns the business process ID. It does not wait for the business process to complete. The user can request the status of the business process at a later

time, using the `getBPStatus()` and `getBPDocument()` methods on the server and supplying the business process ID.

```
String bpID=null;
try {
    bpID=proxy.runBPNowait (NameOfTheBPToInvoke, securityCredentials);
}
catch(Exception e){
    System.out.println("Security, Remote or BusinessProcess exception has
occurred");
    System.out.exit(-1);
}
```

Request Invoked Business Process Status – The `getBPStatus()` method retrieves the status of a business process that was started earlier. You must supply the business process ID. Note that a business process could run for a long time. Account for this when writing your client code.

```
//Get the current status of the invoked business process.
String status=null;
try {
    status=proxy.getBusinessProcessStatus (bpID, securityCredentials);
}
catch(Exception e) {
    System.out.println("Security, Remote or BusinessProcess exception has
occurred");
    System.out.exit(-1);
}
```

Retrieve Resultant Document of the Invoked Business Process – The `getBPDocument()` method retrieves the resultant document from the business process. If the business process completes

successfully, the resultant document is returned to the user as an RMIOPAdapterDocument object; otherwise, null is returned.

```
RMIOPAdapterDocument aDoc=null;
try {
    aDoc=proxy.getBusinessProcessDocument(bpID, securityCredentials);
}
catch(Exception e) {
    System.out.println("Security, Remote exception has occurred");
    System.out.exit(-1);
}
if(aDoc==null) {
    //Business Process must not be complete or BP ID is invalid, etc..
}
else {
    //get the returned document as a string
    String document = new String(aDoc.getBody());
}
}
```

Developing Client Code

Note the following specifications when creating code for the RMI external client method calls.

RMI Adapter .jar Files

The RMI adapter has a .jar file containing adapter class files for developing client code. This .jar file needs to be in the RMI client classpath.

Application Server .jar Files

Each application server has a supporting .jar file or files. Include these files in the client classpath. Each application server packages its .jar files differently. Therefore, the RMI client developer must locate the appropriate .jar file to support an RMI client application.

Report Service

The following table describes the Report service:

System name	ReportService
Graphical Process Modeler (GPM) categories	All Services, System
Description	The Report service produces a report using a pre-defined report configuration. The report is stored in process data with the document name, RPT_<ReportConfigName>.<Format>. For example, if the format is <i>PDF</i> and the Report Configuration name is <i>TestConfig</i> , then the report will be in process data under the document name "RPT_TestConfig.pdf".
Business usage	The Report service enables you to run a report as part of a business process.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	Invoked by a business process. It can be in any step of the business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ SUCCESS – No errors were encountered and a report was stored in process data.◆ ERROR – There was an error encountered as part of the report generation process.
Restrictions	None
Persistence level	System Default
Testing considerations	To test the Report service: <ul style="list-style-type: none">◆ Create and save a report configuration.◆ Supply the format and configuration name to the service. Note: You need to have data in the database to get a meaningful report.

Implementing the Report Service

To implement the Report service, complete the following tasks:

1. Create a Report service configuration. See *Creating a Service Configuration*.
2. Configure the Report service. See *Configuring the Report Service* on page 1186.
3. Use the Report service in a business process.

Configuring the Report Service

To configure the Report service, you must specify field settings in the Graphical Process Modeler (GPM):

Field	Description
Format	The format of the report. Required. Valid values: <ul style="list-style-type: none"> ◆ HTML ◆ PDF ◆ XLS
ReportConfigName	The name of the pre-existing report configuration. Required.

Business Process Example

```
<Report_Svc_BP_Example>
  <sequence>
    <operation name="Report Service">
      <participant name="ReportService"/>
      <output message="Xout">
        <assign to="ConfigName">my_report_configuration</assign>
        <assign to="Format">HTML</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</Report_Svc_BP_Example>
```

Restore Business Process Service

The Restore Business Process service restores business process data from physical media to a Application restored data location where it can be searched and viewed. The following table provides an overview of the Restore Business Process service:

System Name	RestoreService
Graphical Process Modeler (GPM) categories	None
Description	Restores business process data from physical media to a Application restored data location where it can be searched and viewed.
Business usage	This is an internal service that runs from settings in the Archive Manager. It is used to load archived data into a restore area. This data can then be viewed through the appropriate Application user interfaces.
Usage example	This service should not be included in a business process; it is used internally for restore processing.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms Note: You must restore data that was backed up using the same version and patch level as your current installation of Application. Otherwise, the restore will fail.
Related services	Backup Business Process service
Application requirements	Adequate database space must be available to the service for loading archived data into a restore area. However, Documents on Disk (that is, documents that were stored on the disk when the archive was run) are restored to a special location on the disk.
Initiates business processes?	No
Invocation	Invoked by the system business process, RestoreService.bp, for restoring business process data from a backup location.
Business process context considerations	None
Returned status values	None
Restrictions	There must be only one configuration of the Restore Business Process service.
Persistence level	System Default
Testing considerations	None

How the Restore Business Process Service Works

The Restore Business Process service restores business process data that has been backed up and removed from the database and makes it available for viewing. The Restore Business Process service is normally run from the Archive Manager. When restoring archived data, the Restore Business Process service automatically detects if the backed up data is stored in the incremental backup format introduced in Application 4.1.1 or in the older format. It then restores the data to the restore tables accordingly.

Documents on disk that were backed up with their associated table data are restored to a specially created location on the disk. These documents are not restored to the location from where they were first archived. The specially created location keeps restored data out of the live system, and it also allows you to easily delete a restore.


Implementing the Restore Business Process Service

The Restore Business Process service is preconfigured as RestoreService. Do not create additional configurations. If needed, you can edit the RestoreService configuration of the Restore Business Process service in the Application Admin Console. For basic information about editing service configurations, see *Managing Services and Adapters*.

Configuring the Restore Business Process Service

You can specify field settings in Application, using the Admin Console.

Field	Description
Name	Unique and meaningful name for the service configuration. Required. Do not change the preconfigured name.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – Do not include the configuration in a service group at this time.◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.)◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. Note: For more information about service groups, see <i>Managing Services and Adapters</i> .
Restore Thread Pool Size	Maximum number of threads the Restore Business Process service will use for its processing. Required. Default is 8.

Field	Description
Run as User	<p>Applies to the scheduling of the business process.</p> <p>Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list.</p> <p>Valid value is any valid Application user ID.</p> <p>Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select <i>Admin</i> as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>
Use 24 Hour Clock Display	If selected, the service will use the 24-hour clock instead of the default 12-hour clock.
Schedule	<p>Information about scheduling the business process invoked by the Restore Business Process service.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Do not use schedule If this field is selected, the service does not run on a schedule. ◆ Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the service to run at startup. ◆ Run daily Valid values are the hour and minutes at which to run the service, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. ◆ Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. ◆ Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.

SAP Suite Adapter (Build 4300 - Build 4311)

The SAP Suite adapter for SAP R/3 enables you to integrate SAP R/3 versions 3.1i and later with non-SAP applications from vendors, such as PeopleSoft and Oracle, to conduct e-business with trading partners.

The SAP-certified (CA-ALE) SAP Suite adapter enables Application to support platform, data, process, and B2B integration for SAP R/3 by offering the following features:

- SAP inbound and outbound communications processing (ALE (Application Linking and Enabling) and IDOC (Intermediate Documents))

- Synchronous Remote Function Calls (RFCs) and responses

- Synchronous Business Application Programming Interface (BAPI) calls, including transaction handling

- Load balancing for SAP R/3 clusters

- Connection pooling and authentication

The following table provides an overview of the SAP Suite adapter:

System name	SAPSuite
Graphical Process Modeler (GPM) categories	All Services, Applications > ERP
Description	Provides a set of SAP R/3 integration methods.
Business usage	<ul style="list-style-type: none">◆ Near real-time and message-based integration of business processes between non-SAP back-end applications and SAP R/3 systems◆ Support of SAP interfaces: BAPI, synchronous RFC, file-based IDoc RFC (CA-EDI), and ALE/tRfc message-based IDoc (CA-ALE)
Usage example	BAPI example: <ol style="list-style-type: none">1 To determine availability of a certain material in a vendor's stock, you can use a synchronous BAPI call in SAP R/3 and receive the response immediately.2 You can then send a purchase order to buy the material.
Preconfigured?	Partially. You must create a configuration of the SAP Suite adapter and perform minimal configuration to the SAP business processes. The business processes enable SAP inbound and outbound IDoc processing.
Requires third party files?	Yes, SAP Java Connector (JCo) Libraries (most recent version)
Platform availability	All supported Application platforms
Related services	None

Application requirements	SAP connection details (see <i>Before You Begin</i> on page 1205) are required for the SAP Suite adapter to communicate with SAP.
Initiates business processes?	The SAP Suite adapter initiates a business process in Application for the following instances: <ul style="list-style-type: none"> ◆ RFC outbound call ◆ Receipt of an IDoc from SAP R/3 in file-based IDoc RFC mode ◆ Receipt of an IDoc from SAP R/3 in ALE-based IDoc RFC mode
Invocation	The SAP Suite adapter can trigger an outbound business process by configuring the name of the business process. In addition, a business process can start the SAP Suite adapter to perform SAP inbound processing. Data received starts a predetermined business process on an SAP outbound RFC call. For inbound business processes to Application, it is event-driven.
Business process context considerations	For BAPI and synchronous RFC, the SAP Suite adapter creates a primary document in XML format and returns Session or Transaction IDs in the process data. For SAP IDoc outbound, the RFC server creates XML parameters in process data that describe relevant parameters for further processing.
Returned status values	<ul style="list-style-type: none"> ◆ Success ◆ Error – Review advanced status and process data content. Also see <i>Advanced Status Returned by the SAP Suite Adapter</i> on page 1230.
Restrictions	The only supported versions are: <ul style="list-style-type: none"> ◆ SAP R/3 versions 3.1i and later ◆ IDoc versions 3 and 4
Persistence level	None. The persistence is determined by the persistence level of the business processes used for processing inbound and outbound requests.
Testing considerations	Parameter in the Debug Log level and the RFC/JCo trace to get additional debug information on the SAP Suite adapter from library and RFC levels.

How the SAP Suite Adapter Works

To enable a connection to SAP from Application, SAP provides SAP Java™ Connector® (SAP JCo), a free software product for SAP customers which supports connections for the following platforms:

IBM® AIX®
HP-UX11
Linux®
Sun™ Solaris™

Microsoft® Windows®

IBM iSeries

Note: For many platforms, SAP offers 32-bit and 64-bit versions of the JCo libraries, depending on the operating system and the JVM (Java Virtual Machine). Make sure that you install the correct version.

Communicating and Processing IDocs

After installing SAP JCo and the SAP Suite adapter, you configure the adapter and use it in business processes. To communicate with SAP R/3 and your trading partners, and to process IDocs, Application provides several business processes that work together. These business processes use BPML activities, services, and adapters to retrieve documents, perform EDI and IDoc translation, and send documents. These business processes *must* be used together. To implement the business processes, you must perform minimal configuration.

There are two types of IDoc processing: file-based and ALE-based.

For file-based IDoc processing, Application provides the following business processes:

- SAP Inbound IDoc

- SAP Outbound IDoc

- SAP Inbound Delivery

For ALE/IDoc processing, Application provides the following business processes:

- SAP Outbound ALE

- SAP Delivery ALE

SAP Inbound IDoc (SAPInboundIDoc.bp)

SAPInboundIDoc runs after the translation of inbound EDI data to IDoc. Application envelope definitions are associated with SAP routes using the SAP cross-reference configuration. SAP Inbound IDoc inserts the proper routing information into the IDoc control record and transfers the completed IDoc to SAP using the SAP Inbound Delivery business process.

SAP Outbound IDoc (SAPOutboundIDoc.bp)

The SAP Suite adapter retrieves IDocs from SAP R/3. After retrieving IDocs, the SAP Outbound IDoc business process provides end-to-end processing of IDocs. The SAP Outbound IDoc business process enables IDocs to be grouped based on the user-provided EDI envelope definitions. While processing IDocs and preparing them for translation, the SAP Outbound IDoc business process generates status messages that describe processing results. After translation, the SAP Outbound IDocs business process calls the SAP Inbound Delivery business process, which uses FTP to send the status messages back to SAP R/3.

SAP Inbound Delivery (SAPinbDelivery.bp)

Both the SAP Inbound IDoc and SAP Outbound IDoc business processes use the SAP Inbound Delivery business process to send documents. That is:

- After the SAP Inbound IDoc business process performs the final translation on the IDoc (such as adding the routing information to the IDoc control record), the SAP Inbound Delivery business process uses FTP to send the IDocs to SAP R/3.

After the SAP Outbound IDoc business process translates an IDoc received from SAP R/3 to EDI, the SAP Inbound Delivery business process uses FTP to send status messages to SAP R/3.

SAP Outbound ALE (SAPOutboundALE.bp)

The SAP Suite adapter receives IDocs from SAP R/3 using ALE technology. After receiving IDocs, the SAP Outbound ALE business process provides end-to-end processing of IDocs. The SAP Outbound ALE business process enables IDocs to be grouped based on the user-provided EDI envelope definitions. While processing IDocs and preparing them for translation, the SAP Outbound ALE business process generates status messages that describe processing results. After translation, the SAP Outbound ALE business process calls the SAP Delivery ALE business process to send the status messages back to SAP R/3.

SAP Delivery ALE (SAPALEDelivery.bp)

The SAP Outbound ALE business process uses the SAP Delivery ALE business process to send status messages back to SAP R/3 after IDoc to EDI translation is complete.

For more information, see *Implementing the SAP R/3 Business Processes* on page 1214.

Business Scenario

Your company receives a purchase order from a trading partner in EDI format. You need to translate the EDI file to IDoc format and send it using file-based RFC to your back-end SAP system for further processing.

Business Solution Example

The approaches used to solve this business scenario includes the following tasks:

Configure EDI inbound envelopes for the purchase order. You need to define the following information in the appropriate envelopes:

- ◆ The map used to translate the EDI file to IDoc format
- ◆ The predefined SAPInboundIDoc business process as the business process to run after the EDI to IDoc translation is complete

Configure the SAP inbound route that contains the SAP ports and SAP client number of the SAP system that will receive the IDoc. See *Configuring an Inbound Route* on page 1216.

Configure the SAP inbound route cross-reference entry to link the SAP inbound route and EDI inbound envelope for routing the IDoc to the appropriate SAP system. See *Configuring an Inbound Cross-Reference* on page 1219.

Create a business process that includes the EDI Deenvelope service for deenveloping the EDI document.

Configure an instance of the SAP Suite adapter that uses file-based IDoc RFC (see *Example SAP Suite Adapter Configuration* on page 1194).

Update the predefined SAPInbDelivery business process with the appropriate values for sending the IDoc to the SAP system. See *SAP Inbound Delivery (SAPInbDelivery.bp)* on page 1192 for the required updates.

Create an FTP login authentication file required by the FTP server and place the file in the directory where Application is installed. See *Before You Begin* on page 1205.

This business solution example focuses only on the SAP Suite adapter configuration and the SAP Inbound IDoc business process.

Example SAP Suite Adapter Configuration

A sample configuration might look like the following:

Service Settings	
Service Type	SAP Suite Adapter
Description	Example to include in the documentation
System Name	ExampleGPM SAP Suite File Based IDOC RFC
Group	None
SAP Integration Mode	File based IDoc RFC
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	None provided
Gateway Service	None provided
SAP System Number	01
SAP Port	PORT
Map SAP IDoc path to local path	No
Local Path	None provided
Client	999
User	uname
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	No
RFC Server Instances	1
Program ID	Ask_SAP_Administrator
Outbound Process to start	None provided

(Screen 1 of 2)

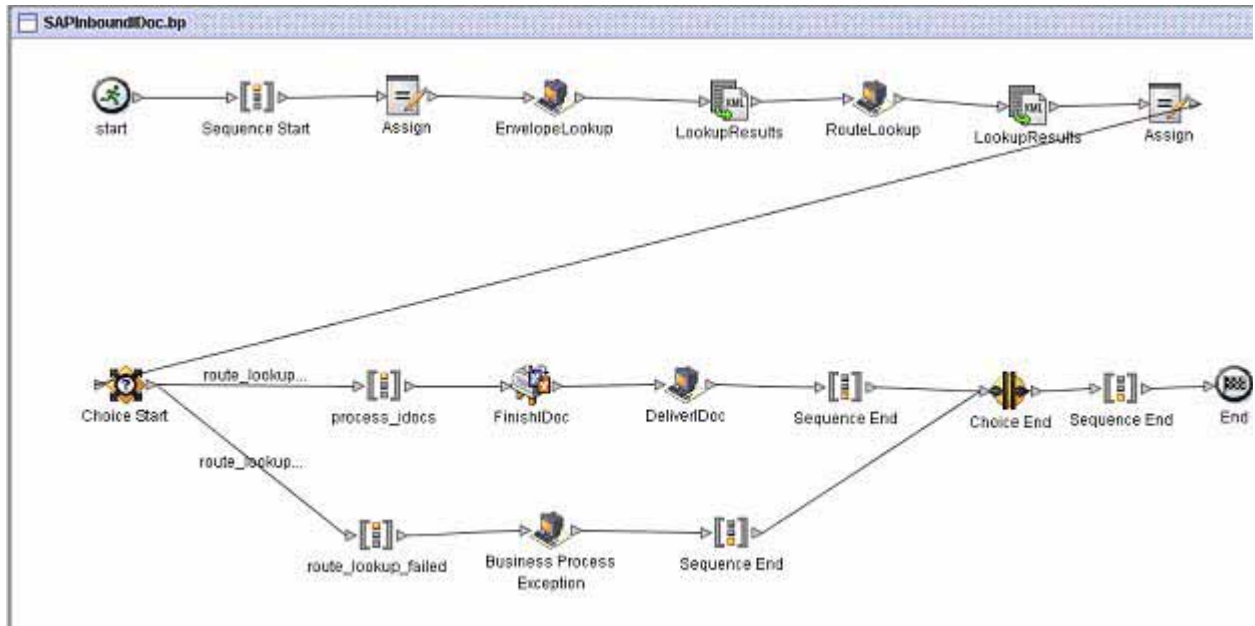
Outbound Process to start	None provided
Outbound Encoding	UTF8
Wait for synchronous RFC Outbound response	No
Response timeout (seconds, 0=unlimited)	0
Delete TIDs automatically	No
Delete TIDs after (days)	30
Register Remote Function Calls (BAPI or RFC Mode only)	None (for IDoc modes or BAPI/RFC Inbound)
Filter online RFC list by	None provided
Use hard max. connections limit (Off= soft limit)	No
Max. Connections	4
Soft Limit Delay Time (seconds)	120
Connection Check Interval (seconds)	30
Connection Idle Time (seconds)	240
Close session after maximum session time	Yes
Max. Session Time (minutes)	60
User	admin

(Screen 2 of 2)

Business Solution Example Business Processes

The following example shows the predefined SAPInboundIDoc business process in the GPM. This business process is specified on the EDI inbound envelope and runs after EDI Deenveloping and after the EDI to

IDoc translation is complete. The translated IDoc is input to the business process and becomes the primary document.



The following example shows the corresponding business process solution using BPML.

```

<process name="SAPInboundIDoc">
  <rule name="route_lookup_results">
    <condition>SAPXRef/IDOCControlData/SAPPORTNAME != &quot;&quot;</condition>
  </rule>
  <sequence name="Sequence Start">
    <assign name="Assign" to="OrigPrimaryDoc" from="PrimaryDocument/@SCIOBJECTID"/>
    <operation name="EnvelopeLookup">
      <participant name="RNJDBCAdapter"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="EnvelopeID/text()"/>
        <assign to="paramtype1">String</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="result_name">SAPXRefEnv</assign>
        <assign to="row_name">EnvelopeName</assign>
        <assign to="sql">select DISTINCT(NAME) from ENVELOPE where ENVELOPE_ID =
?</assign>
        <assign to="." from="*"/>
      </output>
      <input message="inmsg">
        <assign to="." from="*"/>
      </input>
    </operation>
    <operation name="LookupResults">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">SAPXRefEnv</assign>
        <assign to="." from="*"/>
      </output>
      <input message="inmsg">
        <assign to="." from="*"/>
      </input>
    </operation>
    <operation name="RouteLookup">
      <participant name="RNJDBCAdapter"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="SAPXRefEnv/EnvelopeName/NAME/text()"/>
        <assign to="paramtype1">String</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="result_name">SAPXRef</assign>
        <assign to="row_name">IDOCControlData</assign>
        <assign to="sql">select SAPPORTNAME, EDIPORTNAME, CLIENTNUM from
SAP_ROUTE, SAP_XREF where SAP_XREF.ENVELOPE = ? AND SAP_ROUTE.NAME = SAP_XREF.ROUTE</
assign>
        <assign to="." from="*"/>
      </output>
      <input message="inmsg">
        <assign to="." from="*"/>
      </input>
    </operation>
    <operation name="LookupResults">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">SAPXRef</assign>
        <assign to="." from="*"/>
      </output>
      <input message="inmsg">
        <assign to="." from="*"/>
      </input>
    </operation>
  </sequence>
</process>

```

Queries the database for the name of the EDI envelope to retrieve the corresponding SAP route.

Queries the database for the SAP route information based on the EDI envelope name.

```

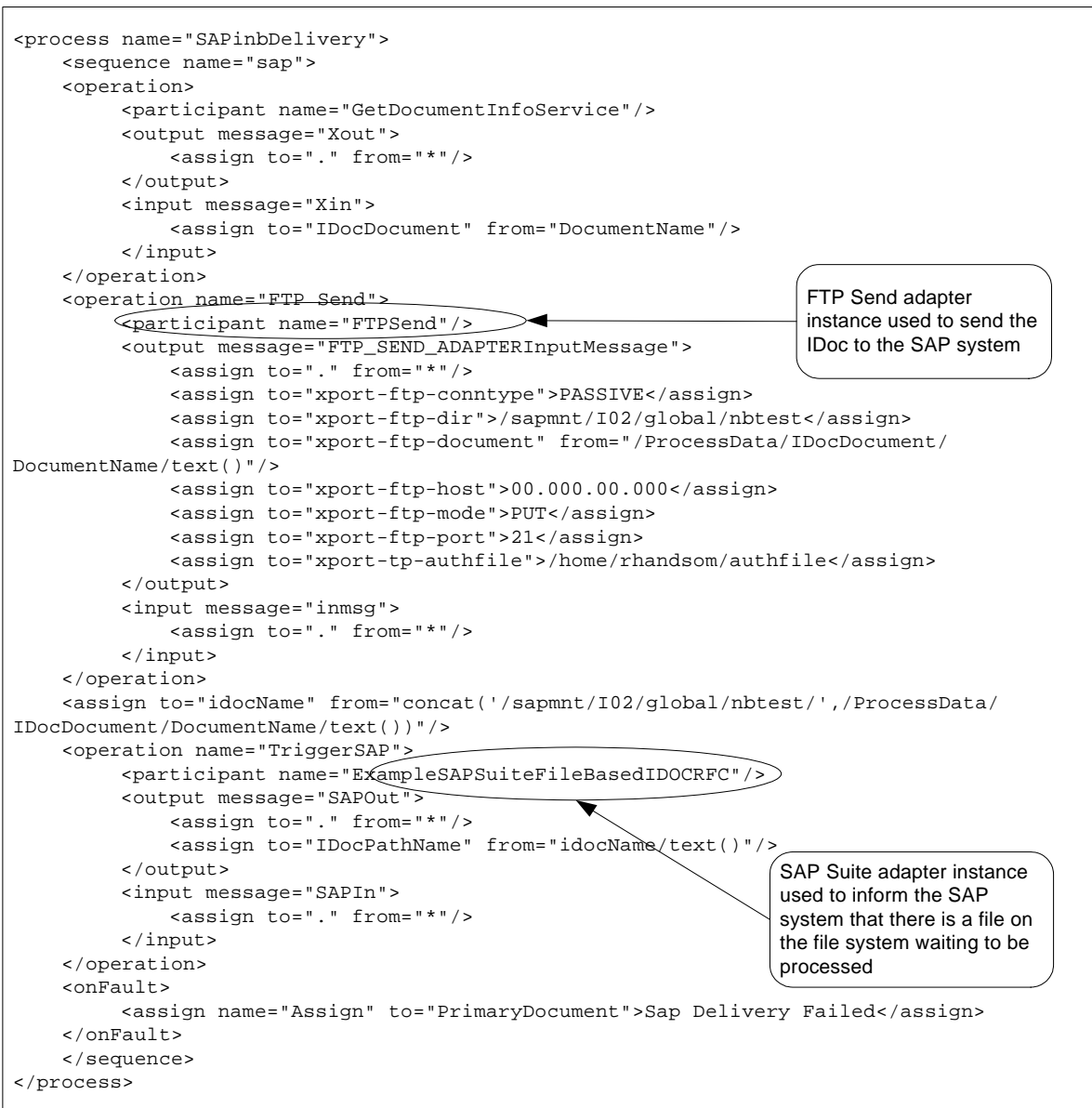
<assign name="Assign" to="PrimaryDocument" from="OrigPrimaryDoc/@SCIOBJECTID"/>
<choice name="Choice Start">
  <select>
    <case ref="route_lookup_results" activity="process_idocs"/>
    <case ref="route_lookup_results" negative="true"
activity="route_lookup_failed"/>
  </select>
  <sequence name="process_idocs">
    <operation name="FinishIDoc">
      <participant name="Translation"/>
      <output message="TranslationTypeInputMessage">
        <assign to="map_name">finishIDoc</assign>
        <assign to="output_to_process_data">NO</assign>
        <assign to="validate_input">NO</assign>
        <assign to="validate_input_against_dtd">NO</assign>
        <assign to="validate_output">NO</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="DeliverIDoc">
      <participant name="InvokeSubProcessService"/>
      <output message="InvokeSubProcessServiceTypeInputMessage">
        <assign to="INVOKE_MODE">ASYNC</assign>
        <assign to="WFD_NAME">SAPinbDelivery</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
  <sequence name="route_lookup_failed">
    <operation name="Business Process Exception">
      <participant name="BPExceptionService"/>
      <output message="BPExceptionServiceTypeInputMessage">
        <assign to="statusReport">Route Lookup Failed</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</choice>
</sequence>
</process>

```

Adds the SAP routing information to the IDoc control record to route the IDoc to the correct SAP system.

Invokes this predefined business process for sending the IDoc to the SAP system.

The following example shows the predefined SAPInbDelivery business process used to send the IDoc to the SAP system.



The following example shows sample output returned to the business process when the IDoc has been successfully received by the SAP system.

```

<EDI_DATA_INCOMING>
<PATHNAME>/sapmnt/I02/directory/orders.dat</PATHNAME>
<PORT>EDI PORT</PORT>
</EDI_DATA_INCOMING>

```

The EDI_DATA_INCOMING element represents the SAP Remote Function Call (RFC) that started. The PATHNAME and PORT elements are the input parameters that were passed into the SAP RFC.

A Basic Status of “Success” and Advanced Status of “None” in the Business Process Modeler indicate that the SAP Suite adapter successfully executed the RFC and that the SAP system has verified the EDI_DC header segment of the IDoc.

Running the Business Solution Example

Assuming that you have completed all the prerequisite requirements for EDI to IDoc document exchanges, you need to perform the following steps to run the business solution example:

1. Run a business process that includes the EDI Envelope service and pass in the EDI file.
2. Verify the results in the Business Process Monitor.

See *Usage Examples* on page 1232 for additional examples of using the SAP Suite adapter.

Implementing the SAP Suite Adapter

To implement the SAP Suite adapter, complete the following tasks:

1. Activate your license for the SAP Suite adapter. For information, see *An Overview of Implementing Services*.
2. Install the SAP Java Connector. For information, see *Installing the SAP Java Connector* on page 1201.
3. Create all of the maps that will be used to translate the IDocs that you are enveloping, and check them into Application.
4. Set up your trading partner information, including EDI interchange, group, and transaction envelopes.
5. Create an SAP Suite adapter configuration. For information, see *Creating an SAP Suite Adapter Configuration* on page 1205.
6. Create business processes or configure/use the following predefined business processes to communicate and implement document processing:

For file-based IDoc processing:

- ◆ SAPInboundIDoc.bp
- ◆ SAPOutboundIDoc.bp
- ◆ SAPInbDelivery.bp

For ALE IDoc processing:

- ◆ SAPOutboundALE.bp
- ◆ SAPALEDelivery.bp

Note: Before you can use your business process, you must check the predefined business processes out of Application. After implementing the predefined business processes, you must check in your business process back to Application.

For more information about implementing SAP business processes, see *Implementing the SAP R/3 Business Processes* on page 1214.

7. Create inbound and outbound routes that describe the key fields in the IDoc that enable Application to route the IDoc to the appropriate trading partner. For information, see *Configuring an SAP R/3 Cross-Reference* on page 1219.
8. Create inbound and outbound cross-references to the routes that enable Application to tie EDI envelopes to their corresponding SAP routes (and vice versa) for routing documents to and from the external trading partners. For information, see *Configuring an SAP R/3 Cross-Reference* on page 1219.

Installing the SAP Java Connector

You must download and install the most recent version of the SAP Java Connector (SAP JCo) before installing the SAP Suite adapter. The SAP JCo contains libraries and packages that support various platforms and enable the SAP Suite adapter to communicate with an SAP system and with Application. After obtaining the SAP JCo, you must make the files available to the host system where Application is installed.

Downloading the SAP Java Connector

To download the SAP Java Connector:

1. Create a user account to access the SAP Service Marketplace offerings.
2. On the same computer where you have installed Application and activated a license for the SAP Suite adapter, download the free SAP JCo from the SAP Service Marketplace Web site at <https://websmp101.sap-ag.de/>.
 - a. Log in to SAP Service Marketplace and access the SAP JCo download software from <http://service.sap.com/connectors>. If necessary, select the **Tools & Services** page to display the download page.
 - b. Download the most recent version of the SAP JCo for your specific platform. The following table lists the specific files for each platform (included in the .tgz file on UNIX or the .zip file on Windows) that the SAP Suite adapter uses in Application:

Note: When transferring files from Windows to UNIX, make sure that you transfer the files in binary mode.

Platform	SAP JAVA Connector Files
IBM AIX	<ul style="list-style-type: none"> ◆ librfccm.o ◆ libsapjcorfc.so ◆ sapjco.jar

Platform	SAP JAVA Connector Files
HP-UX11	<ul style="list-style-type: none"> ◆ librfccm.sl ◆ libsapjcorfc.sl ◆ sapjco.jar <p>Important: For most recent JCo version 2.x.x on HP-UX11: Any messages to stderr in the following format can be ignored: /usr/lib/dld.sl: Can't find path for shared library: dsrlib.sl /usr/lib/dld.sl: No such file or directory</p> <p>These messages have no impact on JCo functionality. This library is used for debugging the sapjco libs (SAPCCMSR). However, if you want to use this library, you can download it from SAP.</p>
Linux	<ul style="list-style-type: none"> ◆ librfccm.so ◆ libsapjcorfc.so ◆ sapjco.jar
Sun Solaris	<ul style="list-style-type: none"> ◆ librfccm.so ◆ libsapjcorfc.so ◆ sapjco.jar
Microsoft Windows	<ul style="list-style-type: none"> ◆ librfc32.dll ◆ sapjco.jar ◆ sapjcorfc.dll <p>Important: For most recent JCo version 2.x.x on Windows: If you are using the SAP JCo version 2.x.x, you must follow the instructions in SAP Note 684106, which explains that the following Microsoft runtime DLLs on your Windows system need to be updated:</p> <ul style="list-style-type: none"> ◆ msvcr71.dll ◆ msvcp71.dll ◆ mfc71.dll ◆ mfc71u.dll <p>The files are attached to the SAP note in an installable archive. Unpack the archive, and then install using R3DLLINS.exe.</p>
IBM iSeries	<ul style="list-style-type: none"> ◆ sapjco.jar ◆ Librfc.savf ◆ Sapjcorfc.savf

Installing on UNIX

To install the SAP Java Connector on UNIX:

1. Complete the procedure *Downloading the SAP Java Connector* on page 1201.
2. Shut down Application if it is running.

3. To set the CLASSPATH and LIBPATH/LD_LIBRARY_PATH, complete the following steps:
 - a. For the operating system-specific libraries, type:


```
install3rdParty.sh sapjco 2_x_x (most recent 2.x.x version) -j directory/*.jar
```
 - b. For the native libraries, type:


```
install3rdParty.sh sapjco 2_x_x (most recent 2.x.x version) -l directory/*.so
```

Notes:

- ◆ Two different switches are used in the previous commands (-j and -l). Make sure you use the correct switch, as shown in the command.
- ◆ lib-extension depends on the operating system you are using. For Sun Solaris, it would be .so, for example. See the table that lists *SAP JAVA Connector Files* on page 1201.
- ◆ If you are using the MySQL database, ensure that the database is stopped after running `install3rdParty.sh`.

4. Restart Application.

Installing on Windows

To install the SAP Java Connector on Windows:

1. Complete the procedure described in *Downloading the SAP Java Connector* on page 1201.
2. Shut down Application if it is running.
3. To add the sapjco.jar and the native libraries to the Windows registry, complete the following steps:

Notes:

- ◆ Two different switches are used in these commands (-j and -l). Make sure that you use the correct switch.
- ◆ If you are using the MySQL database, ensure that the database is stopped after running `install3rdParty.cmd`.

- a. For the operating system-specific libraries, type:


```
install3rdParty.cmd sapjco 2_x_x (most recent 2.x.x version) -j directory\*.jar
```
- b. For the native libraries, type:


```
install3rdParty.cmd sapjco 2_x_x (most recent 2.x.x version) -l directory\*.dll
```

After typing this command, complete the following steps:

- 1) Run `stopWindowsService.cmd`.
 - 2) Run `uninstallWindowsService.cmd`.
 - 3) Close the Command window and re-open it.
 - 4) Run `InstallWindowsService.cmd`.
- c. If you receive a database connection test failure error, in the \bin directory, start MySQL™ using the following command:


```
control_mysql.cmd start
```

4. Restart Application.

Installing on iSeries

To install the SAP Java Connector on iSeries, complete the following steps:

1. Complete the procedure described in *Downloading the SAP Java Connector* on page 1201.
2. From the jar directory for your Application installation, copy **sapjco.jar** into a working directory on your iSeries.

Note: If you use FTP to copy the files, make sure that you transfer the files in binary mode.

3. From the lib directory for your Application installation, copy **LIBRFC.SAVF** and **SAPJCORFC.SAVF** into a working directory on your iSeries.
4. Create a library to store the SAP service programs. From an iSeries command line, type **CRTLIB SAPJCOLIB**. If this library already exists, you can choose another name.
5. Prepare the service programs for the restore by copying them from the IFS to the QSYS side of your iSeries:

```
CPYFRMSTMF FROMSTMF ('/copytodirectory/SAPJCORFC.SAVF')
TOMBR ('/QSYS.LIB/SAPJCOLIB.LIB/SAPJCORFC.FILE')
MBROPT (*REPLACE)
```

and

```
CPYFRMSTMF FROMSTMF ('/copytodirectory/LIBRFC.SAVF')
TOMBR ('/QSYS.LIB/SAPJCOLIB.LIB/LIBRFC.FILE')
MBROPT (*REPLACE)
```

6. Restore the service programs to library SAPJCOLIB by typing:

```
RSTOBJ OBJ(*ALL) SAVLIB(M630SPREL) DEV(*SAVF)
SAVF(SAPJCOLIB/SAPJCORFC) RSTLIB(SAPJCOLIB)
```

and:

```
RSTOBJ OBJ(*ALL) SAVLIB(M630SPREL) DEV(*SAVF)
SAVF(SAPJCOLIB/SAPJCORFC) RSTLIB(SAPJCOLIB)
```

These two commands place the service programs LIBRFC and SAPJCORFC in the library SAPJCOLIB.

7. Edit the job description that you use to start your Application instance. Add the library **SAPJCOLIB** to the initial library list of that job description.
8. Add **SAPJCOLIB** (or a new name, as described in step 4) to the LIBPATH environment variable. Type the following command:

```
ADDENVVAR ENVVAR(LIBPATH) VALUE('/QSYS.LIB/SAPJCOLIB.LIB') LEVEL(*SYS)
```
9. Ensure that your Application instance is not running.
10. From an iSeries command line, type **QSH** and press **Return** to get into Qshell.
11. Change to the bin directory of your Application instance.

12. Type the following command:

```
install3rdParty.sh sapjco lz -j /locationofthesapjco.jarfile/sapjco.jar  
-nodeploy > i3p.log 2> i3p.log
```

13. Upon completion, type:

```
deployer.sh > deployer.log 2> deployer.log
```

14. Once that has completed, verify that the passphrase=password has not been removed from the Application *install_dir/properties/security.properties* file. If it has, add it back.

15. Restart Application.

Creating an SAP Suite Adapter Configuration

Before You Begin

Before you begin to configure the SAP Suite adapter, collect the following information:

SAP Application Server IP address (non-load balanced SAP R/3 system)

SAP Message Server (load balanced SAP R/3 system only)

R/3 Name (load balanced SAP R/3 system only)

SAP Group (load balanced SAP R/3 system only)

SAP Gateway Host IP address (if applicable, required for SAP outbound processing)

SAP Gateway Service (if applicable, required for SAP outbound processing)

SAP System Number

SAP Port (Mode File based IDoc RFC only)

SAP Client Number

SAP Program ID (SAP outbound processing only)

Username for logging into the SAP Application Server

Password for logging into the SAP Application Server

Name of the business process to start (SAP outbound processing only)

Name of the Remote Function Call(s) to register (SAP RFC synchronous requests and BAPI only.
Required for SAP outbound processing only.)

See *Configuring the SAP Suite Adapter* on page 1206 for a description of the parameters used to define this information.

In addition, for file-based IDoc processing, you need to create an FTP login authentication file (required by the FTP server) and place the file in the directory where Application is installed. The file must contain the following values:

```
username=sapusername
```

```
password=sappassword
```

Configuring the SAP Suite Adapter

Whether you plan to create a business process that includes the SAP Suite adapter or use the predefined business processes, you must create a service configuration of the SAP Suite adapter. For more information, see *Managing Services and Adapters*.

To create and enable a configuration of the SAP Suite adapter, use the following table to configure the parameters:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select a Service Group to associate with this adapter. Valid values: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. For more information on Service Groups see chapter 1, <i>Managing Services and Adapters</i> .
SAP Integration Mode (Mode)	Integration mode that enables Application to communicate with an SAP R/3 system and with transfer documents. Required. Valid values: <ul style="list-style-type: none">◆ Synchronous BAPI – Enable integrated activities using SAP R/3 business objects and their BAPIs, which are processed in synchronous mode.◆ Synchronous RFC – Trigger direct and synchronous RFCs that start SAP R/3 function modules or receive synchronous RFC calls from SAP R/3.◆ File-based IDoc RFC – Transfer documents using SAP R/3 IDoc technology when communicating with SAP R/3 using RFC.◆ ALE-based IDoc – Transfer documents using SAP R/3 IDoc technology when communicating with SAP R/3 using ALE.
SAP system is load balanced (LoadBalanced)	Whether Application is connecting to a single SAP R/3 system or cluster of SAP R/3 systems. Required. Valid values: <ul style="list-style-type: none">◆ Yes – Connect to a cluster of SAP R/3 systems.◆ No – Connect to a single SAP R/3 system.

The integration mode and load-balancing determine which of the following parameters display for configuration. Refer to the following sections to determine the information that you must provide to continue creating the SAP Suite adapter service configuration:

Synchronous BAPI on page 1207

Synchronous RFC on page 1208

File-based IDoc RFC on page 1208

ALE-based IDoc RFC on page 1209

User Properties on page 1210

RFC Server Configuration (Outbound) on page 1211

Connection Pool Settings on page 1212

Application User Settings on page 1213

Synchronous BAPI

The following table describes the fields to configure for synchronous BAPI:

Field	Description
Load Balanced	
SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the SAP R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Not Load Balanced	
SAP Application Server	Name of the server for the SAP R/3 application. Required.
Gateway Host (GWhost)	Host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (GWserv)	Gateway service of the SAP R/3 system. Optional.
SAP System Number	Number of the SAP R/3 system. Required.

Synchronous RFC

The following table describes the fields to configure for synchronous RFC:

Field	Description
Load Balanced	
SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the SAP R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Gateway Host (GWhost)	Host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (GWserv)	Gateway service of the SAP R/3 system. Optional.
Not Load Balanced	
SAP Application Server	Name of the server for the SAP R/3 application. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use your IP address. Optional.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.
SAP System Number	Number of the SAP R/3 system. Required.

File-based IDoc RFC

The following table describes the fields to configure for file-based IDoc RFC:

Field	Description
Not Load Balanced	
SAP Application Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.
SAP System Number	Number of the SAP R/3 system. Required.

Field	Description
SAP Port (Port)	Port or communication type that enables the SAP R/3 system to communicate with external systems during electronic data interchange. For example, for EDI subsystems that read IDocs in the form of sequential files, the SAP R/3 port is File. Required.
Map SAP IDoc path to local path	Enable Application to map the local directory path to the IDoc on the SAP R/3 system.
Local Path (MapPath)	Local directory path to the IDoc on the SAP R/3 system. Required if you selected the Map SAP IDoc path to local path check box.
Load Balanced	
SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Optional.
SAP Port (Port)	Port or communication type that enables the SAP R/3 system to communicate with external systems during electronic data interchange. For example, for EDI subsystems that read IDocs in the form of sequential files, the SAP R/3 port is File. Required.
Map SAP IDoc path to local path	Enable Application to map the local directory path to the IDoc on the SAP R/3 system.
Local Path (MapPath)	Local directory path to the IDoc on the SAP R/3 system. Required if you selected the Map SAP IDoc path to local path check box.

ALE-based IDoc RFC

The following table describes the fields to configure for ALE-based IDoc RFC:

Field	Description
Not Load Balanced	
SAP Application Server	Name that identifies the application server for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Required.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.

Field	Description
SAP Version (SAPVersion)	Version of the SAP R/3 system to control IDoc version. Required. Valid values are: <ul style="list-style-type: none"> ◆ 3 – IDoc version used in SAP releases 3.x (EDI_DC) ◆ 4 – IDoc version used in SAP releases 4.x (EDI_DC40)
Check box Retry Sending IDocs	Check if SAP Suite adapter should automatically retry to deliver the IDoc to the SAP system.
Max Retries	Sets how often the retry should take place. Setting the value to 0 causes the SAP Suite adapter to retry infinitely.
Retry Sending Interval	Delay in seconds between two retries.

Load Balanced

SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the SAP R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.
SAP Version (SAPVersion)	Version of the SAP R/3 system. Required.
Check box	Check if SAP Suite adapter should automatically retry to deliver the IDoc to the SAP R/3 system.
Max Retries	Sets how often the retry should take place. Setting the value to 0 causes the SAP Suite adapter to retry infinitely.
Retry Sending Interval	Delay in seconds between two retries.

User Properties

The following table describes the fields to configure user properties:

Field	Description
Client (Client)	Name of the client in the SAP R/3 system. Required.
User (User)	Name of the authorized user of the SAP R/3 account. Required.
Password (Passwd)	Password of the authorized user of the SAP R/3 account. Required.

Field	Description
Language (Lang)	Language used by the SAP R/3 account. Required. Valid value is EN (English).
Code Page (Codepage)	Code page of this SAP R/3 system—for example, 1100 for English and most European languages. Optional.

RFC Server Configuration (Outbound)

The following table describes the fields to configure RFC properties:

Field	Description
Start RFC Server automatically	Start the local RFC server automatically. The RFC server is required for SAP R/3 outbound.
RFC Server Instances	Number of RFC server instances listening for connections from the SAP R/3 system.
Program ID (Program ID)	The SAP system program ID used to tie the SAP Suite adapter instance to the correct SAP system for handling outbound requests.
Outbound Process to Start	Select the name of the business process that Application should run after receiving an IDoc from SAP R/3.
Outbound Encoding	Set the character encoding for the outbound document.
Wait for synchronous RFC Outbound response	Enables support for synchronous responses in case of outbound SAP RFC calls.
Response Timeout	After this timeout (in seconds), the RFC server closes the connection to the SAP system. After this timeout, if Application wants to send a response back to SAP, an error is returned.
Delete TIDs Automatically	Indicates whether transaction IDs (TIDs) that have been already used should be deleted after the time configured in the Delete TIDs after (days) field. Note: TIDs are used for ALE processing only.
Delete TIDs after (days)	TIDs older than <i>n</i> days are deleted.

Field	Description
Register Remote Function	<p>This applies to BAPI or RFC Outbound mode only. To be able to handle outbound BAPI or RFC calls, the call structures have to be registered in Application.</p> <p>Options are:</p> <ul style="list-style-type: none"> ◆ Select RFCs online from RFC list Application establishes a connection to SAP and retrieves the list of available RFC or BAPI functions. ◆ Enter RFCs offline A comma-separated list of RFCs can be entered offline; no connection to the SAP system is established. ◆ None (for IDoc modes or BAPI/RFC Inbound): No functions need to be registered.
Checkbox SAP R/3 is a UNICODE system	<p>To be able to interface with Unicode SAP systems, you must enable the check box.</p> <p>Note: This option makes it unnecessary to add the line "sap.(instancename).Unicode = 1" to the sap.properties file, which was necessary in earlier versions of the SAP Suite adapter:</p>

Note: For Unicode SAP systems:

To be able to interface with Unicode SAP systems, you must set the Unicode option for your RFC destination in transaction SM59 on the SAP system (please contact the SAP system administrator for more information). You can find this option on the **Special Options** tab, in the **Character Width in Target System** section.

Connection Pool Settings

There are two types of connections to an SAP system: connections (physical) and sessions (logical).

Connections: Physical connections to an SAP system. Connections can be in either closed or open status.

Sessions: Logical connections to an SAP system. Each session object contains a connection object. A session can be in either used, unused, or called status.

The following table contains the settings used to configure a connection pool:

Field	Description
Use hard max connections limit	Check this box if the number of open parallel connections should never exceed the number of configured maximum connections as configured in Max Connections (Off = soft limit).
Max Connections	Maximum number of open parallel sessions.
Soft Limit Delay Time (seconds)	After this time, if all max sessions are still used and a call is waiting to be executed, an additional session is created.
Connection Check Interval (seconds)	Interval in which the session and connection state is checked by the adapter.

Field	Description
Connection Idle Time (seconds)	If a session is unused for this period of time, the connection is closed.
Close session after maximum session time	Whether the adapter should close the connections after the Max Session Time (see next field). Select this option to prevent connections from staying opened indefinitely.
Max Session Time (minutes)	Maximum time after a session is not processed (if enabled).

Note: If invalid data (like ABC or 13 . 45) is entered in a pool setting, the setting uses its default value.

Application User Settings

The following table contains the user setting for the service configuration:

Field	Description
User ID	Name of the user to associate with the business process that includes this service configuration.

You are now ready to complete either of the following tasks:

- Create a business process that includes the SAP Suite adapter.
- Configure SAP R/3 business processes for implementation (see the next section).

Editing the SAP Suite Adapter Configuration Parameters

Upon installation, the connection pool, RFC server, and trace parameters that enable connection to an SAP system are preconfigured.

To edit these parameters:

1. From the **Administration** menu, select **Deployment > Services > Installation/Setup**.
2. Under List, select **S** next to Alphabetically and click **Go!**

Note: Under List for Service Status, Installed (default value) should display.

3. From the list of services, select **edit** next to SAP Suite adapter.
4. Using the following table, edit the fields as appropriate:

Field	Description
Max. Startup Delay (MaxStartupDelay)	Maximum number of startup connections to the SAP R/3 system. Required. The default value is 60 seconds.

Field	Description
RFC Trace (RFCTrace)	RFC trace processing information about connections in the pool. Valid values: <ul style="list-style-type: none"> ◆ On – Trace the connection. ◆ Off – Do not trace the connection.
JCO Trace Level (JCOTraceLevel)	JCo trace processing information level. Valid values are 0 to 6 (6 is the maximum level).
JCO Trace Path (JCOTracePath)	Directory for the JCo trace output files. If “.” is used, the trace files will be located in the application server installation directory.

5. Click **Next** and review the parameters that you edited.
6. Click **Finish** to update Application.

Implementing the SAP R/3 Business Processes

To implement the business processes for SAP R/3, you need to perform the minimal configurations for the SAP Outbound IDoc, SAP Inbound Delivery, and SAP Delivery ALE business processes. The SAP Inbound IDoc and SAP Outbound ALE business processes require no additional setup.

To implement the business processes:

1. Check out the following predefined business processes (.bp files) from Application:
 - ◆ SAPInbDelivery.bp (for file-based IDoc)
 - ◆ SAPOutboundIDoc.bp (for file-based IDoc)
 - ◆ SAP Delivery ALE (for ALE IDoc)
2. After checking out the business processes, use the element editors in the GPM to display the configuration parameters within each business process.

Refer to the following sections to determine the information that you must provide to implement the business processes.

SAP Outbound IDoc (SAPOutboundIDoc.bp)

The following table provides the parameters to define for the preconfigured services, as appropriate:

Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the FTP Send adapter has entered the retirement process in Application and will be

replaced with the FTP Client Adapter with related services. For more information about the retirement process, see *Retiring and Removed Services and Adapters*.

Service/Adapter Configuration Instance Name	Parameter	Description
FTPSend (FTP Send Adapter)	xport-ftp-document	Name of IDoc that Application retrieves from SAP R/3. Required.
	xport-ftp-host	IP address or host name of the external trading partner host system. Valid values are valid IP addresses and host names. Required.
	xport-tp-authfile	Authentication file containing the user name, password, and passphrase. Valid value is the file name. Required if passphrase is used.
DocumentExtractionService (Document Extraction Service)	BatchLikeDocuments	Whether to split the IDocs that are extracted into batches. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes – Group IDocs into files based on Sender ID, Receiver ID, and Acceptor Lookup Alias. ◆ No – Split out each IDoc individually, regardless of Sender ID, Receiver ID, and Acceptor Lookup Alias values.
	DocExtractMapList	Name of the map to extract documents from a single batch file. Required if splitting IDocs into batches.
FS_WriteEDI (File System Adapter)	assignedFilename	Unique file name used to overwrite the file name created by the business process. If the file name is not unique, the previous file with the same name is overwritten the next time it runs. Valid value is any valid file name. Required.
	extractionFolder	Any folder or subfolder on the same computer where Application is installed and where it extracts (writes) data as part of a business process. Required.

SAP Inbound Delivery (SAPInbDelivery.bp)

The following table provides the parameters to define for the preconfigured services, as appropriate:

Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the FTP Send adapter has entered the retirement process in Application and will be

replaced with the FTP Client Adapter with related services. For more information about the retirement process, see *Retiring and Removed Services and Adapters*.

Adapter Configuration Instance Name	Parameter	Description
FTPSend (FTP Send Adapter)	xport-ftp-dir	Folder name or mailbox ID of an external trading partner. Required.
	xport-ftp-document	Document that the trading partner within Application retrieves from the remote trading partner. Optional.
	xport-ftp-host	IP address or host name of the external trading partner host system. Valid values are valid IP addresses and host names. Required.
	xport-tp-authfile	Authentication file containing the user name, password, and passphrase. Valid value is the file name. Required if passphrase is used.
SapSuite (SAP Suite Adapter)		Configure the parameters for this preconfigured instance of the SAP Suite adapter or specify your custom SAP Suite adapter configuration instance. See <i>Creating an SAP Suite Adapter Configuration</i> on page 1205.

SAP Delivery ALE (SAPALEDelivery.bp)

The following table provides the parameters to define for the preconfigured services, as appropriate:

Adapter Configuration Instance Name	Parameter	Description
SapSuite (SAP Suite Adapter)		Configure the parameters for this preconfigured instance of the SAP Suite adapter or specify your custom SAP Suite adapter configuration instance. See <i>Creating an SAP Suite Adapter Configuration</i> on page 1205.

You are now ready to check in the business processes to Application.

Configuring an SAP R/3 Route

Application uses SAP routes to determine how to route IDocs to and from external trading partners. When creating the inbound routes, indicate which key fields in the IDoc EDI_DC control record are used to identify the IDoc.

Configuring an Inbound Route

To configure an inbound route:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Routes**.
2. Under Create, next to **New Inbound Route**, click **Go!**
3. Complete the fields in the following table, as appropriate:

Field	Description
Route Name	Unique, meaningful name for the inbound route. Required.
SAP Direction	Direction of this route. This value is read-only and is set to inbound. No configuration necessary.
SAP IDOC Version	Version of the IDoc that you are exchanging with SAP R/3. Required.
EDI Port Name	Name of the port that enables communication between Application and an SAP EDI subsystem. The EDI subsystem triggers the SAP Suite adapter when sending IDocs to Application. Application triggers the EDI subsystem when sending IDocs to SAP. Required. Note: Both the SAP port name and EDI port name are used to route the IDoc to the correct trading partner. These values, along with the SAP client number, are added to the IDoc control record during the final translation of the IDoc in the SAPInboundIDoc business process.
SAP Port Name	Name of the SAP internal port that enables communication between the EDI subsystem and the SAP system when sending and receiving IDocs. Required. Note: Both the SAP port name and EDI port name are used to route the IDoc to the correct trading partner. These values, along with the SAP client number, are added to the IDoc control record during the final translation of the IDoc in the SAPInboundIDoc business process.
SAP Client Number	SAP R/3 client number. Required.

4. Click **Next** and review your configuration settings.
5. Click **Finish** to add the inbound SAP R/3 route to Application.

Configuring an Outbound Route

To configure an outbound route:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Routes**.
2. Under Create, next to **New Outbound Route**, click **Go!**

3. Complete the fields in the following table, as appropriate:

Field	Description
Route Name	Unique, meaningful name for the outbound route. Required.
SAP Direction	Direction of this route. This value is read-only and is set to outbound. No configuration is necessary.
SAP IDOC Version	Version of the IDoc that you are exchanging with SAP R/3. Required.
EDI Port Name	<p>Name of the port that enables communication between Application and an SAP EDI subsystem. The EDI subsystem triggers the SAP Suite adapter when sending IDocs to Application. Application triggers the EDI subsystem when sending IDocs to SAP. Required.</p> <p>In addition, the EDI port name is included in the IDoc header and used along with the SAP port name and Acceptor Lookup Alias defined for the EDI envelope to identify the inner-most EDI envelope (which contains the translation map) to apply to the document. You must specify the EDI port name as the Receiver ID for the inner-most EDI envelope (for example, ST SE envelope).</p>
SAP Port Name	<p>Name of the SAP internal port that enables communication between the EDI subsystem and the SAP system when sending and receiving IDocs. Required.</p> <p>In addition, the SAP port name is included in the IDoc header and used along with the EDI port name and Acceptor Lookup Alias defined for the EDI envelope to identify the inner-most EDI envelope (which contains the translation map) to apply to the document. You must specify the SAP port name as the Sender ID for the inner-most EDI envelope (for example, ST SE envelope).</p>
SAP Partner Keys	<p>The key fields from the IDoc that Application uses to identify the IDoc. Required. Select all the key fields that apply.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ MESCOD – Logical message variant ◆ MESFCT – Logical message function ◆ MESTYP – Logical message type ◆ RCVPFC – Partner function of receiver ◆ RCVPRN – Partner number of receiver ◆ RCVPRT – Partner type of receiver ◆ SNDPRN – Partner number of the sender ◆ SNDPRT – Partner type of the sender ◆ STDMES – EDI message type ◆ TEST – Test option

4. Click **Next** and review your configuration settings.

5. Click **Finish** to add the outbound cross-reference to Application.

Configuring an SAP R/3 Cross-Reference

To enable Application to process inbound and outbound IDocs and translate them to and from EDI, you must specify SAP cross-references to look up the EDI envelope associated with the SAP route defined for the IDoc.

Configuring an Inbound Cross-Reference

To configure an inbound cross-reference:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Route X-REF**.
2. Under Create, next to New Inbound X-REF, click **Go!**
3. Complete the fields in the following table, as appropriate:

Field	Description
Route Selection	Name of the inbound route to which you are creating this cross-reference. Required.
Envelope Selection	EDI envelope that you have previously created to translate the key field or fields that you specified when creating an inbound route. Required. Note: Specify the inner-most EDI envelope in this field.

4. Click **Next** and review your configuration settings.
5. Click **Finish** to add the inbound cross-reference to Application.

Configuring an Outbound Cross-reference

To configure an outbound cross-reference:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Route X-REF**.
2. Under Create, next to New Outbound X-REF, click **Go!**
3. Complete the fields in the following table, as appropriate:

Field	Description
Selection Route	Name of the outbound route to which you are creating this cross-reference. Required.
Select Envelope	EDI envelope that you have previously created to translate the key field or fields that you specified when creating an outbound route. Required. Note: Specify the inner-most EDI envelope in this field.

4. Click **Next** and review your configuration settings, and then click **Finish** to add the outbound cross-reference to Application.

Configuring for Load Balancing

If the SAP system is load-balanced (“SAP system is loadbalanced” = Yes in the SAP Suite adapter configuration), you must configure the /etc/services file as shown in the following table. The /etc/services file is used to map port numbers to service names.

The location of this file depends on your host system (UNIX/Linux, Windows, or iSeries). In UNIX, the file is at /etc/services. In Windows, the file is at c:\WINDOWS\system32\drivers\etc\services. In iSeries, use the iSeries Transaction WRKSRVTBLE to edit this information.

Service Type	Port	Service Name
sapmsS01	3600/tcp	SAP System Message Port
sapdp00	3200/tcp	SAP System Dispatcher Port
sapdp00s	4700/tcp	SAP System Dispatcher Security Port
sapgw00	3300/tcp	SAP System Gateway Central Instance Port
sapgw00s	4800/tcp	SAP System Gateway Security Port

For the sapmsXXX service type, XXX is the SAP System ID. For example, if the SAP System ID is E01, the entry should read as follows:

```
sapmsE01 35YY/tcp #SAP System Message Port
```

YY is the SAP System Number.

Retrieving IDoc and Schema Format Descriptions

IDoc and schema format descriptions can be obtained either by using the new SapSuiteBuilder interface or with the command line-based tool, CmdlineSAPSuiteBuilder.

Retrieving Descriptions Using the SAP Suite Builder Interface

To retrieve IDoc and schema format descriptions using the SAP Suite Builder interface:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Suite Builder**.
2. Enter the parameters required to access and log in to the SAP system.
Application connects to the SAP system and retrieves a list of available IDocs, BAPIs (objects and methods), or RFCs.
3. Select the required format descriptions.
4. Click **Finish** on the summary screen.
5. Download the formats as .ddf or .xsd files.
6. Import these format descriptions into the Application Map Editor.

Note: The **IDoc Meta Data Builder > Generation Parameters** page now includes the parameter **Generate IDoc version in records**. If enabled, the generated IDoc Record tag includes the segment definition name (*SEGMENTDEF*, <segment_type><segment_version>). If not enabled, the

generated IDoc Record tag includes the segment type name only (*SEGMENTTYP*, *<segment_type>*). This was the default in earlier versions of the SAP Suite Builder.

Retrieving Descriptions Using the SAP Command Line Interface

This section contains information about retrieving IDoc and schema format descriptions using the SAP Command Line interface. The `CmdlineSAPSuiteBuilder` is a command line-based tool that is installed when the SAP Suite adapter is installed. The `CmdlineSAPSuiteBuilder` enables a connection to an SAP R/3 system and delivers a particular IDoctype or schema (or its extension) in the form of a data definition format (.ddf) or schema (.xsd) file (see *DDF in create mode* on page 1223 or *XSD in create mode* on page 1224). In List mode, the `CMDLineSAPSuiteBuilder` delivers a list of either IDocs or schemas from the SAP R/3 system to which it is connected. You can then use the .ddf and the .xsd in the Application Map Editor to define your mapping requirements.

To use the utility, you should have a basic knowledge of:

- Types of IDocs (and their extensions) and RFCs (RFMs) in an SAP R/3 system
- Data formats (.ddf and .xsd)
- Data mapping concepts
- Application Map Editor

Requirements

Based on SAP R/3 requirements for retrieving IDocs, the `CMDlineSAPSuiteBuilder` utility needs particular RFC function calls, including `IDOCTYPE_READ_COMPLETE` and `IDOC_RECORD_READ`. Before running the utility, confirm that the necessary RFC function calls are available.

Note: The `IDOCTYPE_READ_COMPLETE` and `IDOC_RECORD_READ` RFC function calls are available in SAP release 4.5A (as standard) or 4.0B (with the additional support packages `SAPKH40B42` and `SAPKH40B48`).

Running the CmdlineSAPSuiteBuilder Utility

To run the `CmdlineSAPSuiteBuilder` utility, type one of the following commands:

UNIX:

```
bin/runSAPClass.sh schemaGenerator.CmdlineSAPSuiteBuilder <arguments>
```

Windows:

arguments

The `bin/runSAPClass.sh` or `bin\runSAPClass.cmd` script is a generic script that enables you to call any class (that has a main method) inside the SAP-related Application environment (that is, the path begins with `com.sterlingcommerce.Woodstock.services.sapsuite`). Therefore, the script expects, as a first parameter, the name of the Javaclass call (in this case, `schemaGenerator.CmdlineSAPSuiteBuilder`). The script may also call another external class file inside the `sapsuite` environment.

To see the syntax and usage information, type the command without *<arguments>*, with the following result:

|| -l list IDOCs or RFCs.

<DDFBuilder-options> - M)andatory or O)ptional - in create mode are:

-e <Extension> O (CIMtype, e.g. ZORDERS01)
-i <IDOCType> M (Basicitype e.g. ORDERS01)
-n O (no IDoc version in records will be generated)
-o <outputfilename> O (name of DDF-file to be generated)
-p <propertyfilename> M (file with SAP specific properties)
-s <Segment_Release> O (SAP Release (3 alphanum. chars, e.g. 40B))
-v <Record Type Vers> O (allowed values: 2 or 3 (default))

<DDFBuilder-options> - M)andatory or O)ptional - in list mode are:

-f <Filterargument> M (search for a list of IDCOS e.g. ORDERS*)
-o <outputfilename> O (name of listfile that keeps the found elements)
-p <propertyfilename> M (file with SAP specific properties)

-s <Segment_Release> O (SAP Release (3 alphanumeric chars, e.g. 40B))

<SchemaBuilder-options> - M)andatory or O)ptional - in create mode are:

-o <outputfilename> O (name of xsd-file to be generated)
-p <propertyfilename> M (file with SAP specific properties)
-r <RFCname> M (RFCname whose Schema will be extracted)

<SchemaBuilder-options> - M)andatory or O)ptional - in list mode are:

-p <propertyfilename> M (file with SAP specific properties)
-f <Filterargument> M (search for a list of BO's, BAPI's or RFC's depending on searchtype)
-o <outputfilename> O (name of listfile that keeps the found elements)
-t <searchtype> M (allowed values: BO or BAPI or RFC)
-a <additional arg> O (used in case of BAPI (M) or RFC (O))

There is always a create mode or a list mode for DDF and for XSD. The following examples illustrate how to use the different modes of the CmdlineSAPSuiteBuilder utility:

DDF in create mode

<arguments> =

```
DDF -c -i ORDERS01 -s 45A -v 3 -p SAPreadIdoc.properties -o orders01.ddf
```

Create the ddf file **orders01.ddf** that contains the IDoc structure of the IDOCTYPE ORDERS01 of segment release 45A and record type version 3. The corresponding SAP-related host and user information are read from the file SAPreadIdoc.properties.

The CmdlineSAPSuiteBuilder utility requests the password for the SAP username (field User) as configured in the property file (this is the case in every mode).

In addition, the CmdlineSAPSuiteBuilder program writes to a log file whose name is specified with the token saplogger.logfilename in the property file log.properties. Its amount of output ranges from only fatal messages to all messages (NONE, FATAL, ERROR, WARN, TIMING, INFO, DEBUG, ALL). Its value depends on the token saplogger.loglevel in the same property file log.properties that is common to the SAP Suite adapter package. This logging mechanism is in every mode.

DDF in list mode

<arguments> =

```
DDF -l -f ORDERS -s 45A -p SAPreadIdoc.properties
```

Deliver to stdout a list of all IDOCTYPES (and/or extensions) beginning with the string ORDERS.

XSD in create mode

<arguments> =

```
XSD -c -r BAPI_MATERIAL_AVAILABILITY -p SAPreadIdoc.properties -o  
BAPI_Material_Availibility.xsd
```

Create an XML schema from the RFC BAPI_MATERIAL_AVAILABILITY that, in this case, stands for a particular BAPI of the BusinessObject MATERIAL in the file BAPI_Material_Availibility.xsd. SAP-related properties are read from the file SAPreadIdoc.properties.

XSD in list mode

Using this, it is possible to filter for RFCs, BAPIs, or BOs, depending on the search type used:

Listing of RFCs

<arguments> =

```
XSD -l -t RFC -f BAPI_SALESORDER_CREATEFROMDAT -p SAPreadIdoc.properties
```

List all RFCs beginning with BAPI_SALESORDER_CREATEFROMDAT.

Listing of BOs

<arguments> =

```
XSD -l -t BO -f Mat -p SAPreadIdoc.properties
```

List all BusinessObjects beginning with the string “Mat.”

Listing of BAPIs

<arguments> =

```
XSD -l -t BAPI -a MATERIAL -f Get -p SAPreadIdoc.properties
```

List all BAPIs beginning with the string “Get” of the particular BusinessObject Material. Note that BAPIs can only be listed for one particular BusinessObject.

Property File Used by CmdlineSAPSuiteBuilder

The property file must be customized before using the utility. Each line in this file should have the following format:

<fieldname> = <value>

The following table describes the possible field names and their default values in the property file:

Note: All mandatory fields have no default value and therefore have to be set with the correct value. The range for the JCoTraceLevel is undocumented but the lowest level is 0 (no tracing).

Field Name	Default Value	Non LB, M or O/LB, M or O	Notes/Used Datatype
Gwhost		O	IP address or hostname
Gwserv		O	

Field Name	Default Value	Non LB, M or O/LB, M or O	Notes/Used Datatype
Client		M	Three-digit number
User		M	Maximum of 12 characters
Lang	EN	O	Exactly two characters
OutboundEncoding	UTF-8	O	Encoding used during generation of the DDF/schema definitions
Codepage	1100	O	Four-digit number
LoadBalanced	LB_OFF	O	If LB_OFF, the fields Ashost and Sysnr are mandatory. If LB_ON, the fields Mshost and R3name are mandatory.
Ashost		M / -	If not LoadBalanced: IP address or hostname
Sysnr		M / -	If not LoadBalanced: Number
Mshost		- / M	IP address of message server; if you used field name Ashost, this field is not used.
R3name		- / M	SAPSysID
RFCTrace	RFCTRACE_OFF	O	switch on: RFCTRACE_ON
JCoTraceLevel	0	O	Range: 0 - 9
JCoTracePath	.	O	

Error Messages

When the CmdlineSAPSuiteBuilder utility detects errors, the utility writes the errors to either stdout or to the logfile associated with saplogger.logfilename. Typically, errors are written to stdout only if during command-line parsing something was wrong. At this stage, the name of the logfile is still not known.

Business Process Definition Parameters –Transactions

The following table describes the business process parameters for transactions:

Parameter	Data Type/ HTML Type/ Validator/ Size/MaxSize	Value/ Default	Description
KeepSessionOpen	String/text/ Number	0,1	BAPI only: Causes the SAP Suite adapter to leave the connection open after executing the BAPI.
SessionID	String/text/	<sid>	BAPI only: Causes the SAP Suite adapter to connect to the SAP system using the session <sid> that was opened in a previous call. For information, see KeepSessionOpen (above).
BapiCommitWait	String/text	0, 1	BAPI only: BapiCommitWait is an optional parameter for a BapiCommit call. Causes the SAP Suite adapter to commit all pending calls and wait until the commit returns. (1=Wait, 0=Do not wait)
BapiCommit		0, 1	BAPI only: Causes the SAP Suite adapter to commit all pending calls without waiting. The result is returned in a primary document.
BapiRollback		0, 1	BAPI only: Causes the SAP Suite adapter to roll back all pending calls. The result is returned in a primary document.
RFCModuleName	String/text		RFC/BAPI only: The name of the remote function module or BAPI RFC to be run.
Encoding	String/text	UTF-8	Specifies the character set.
CreateTID	String/text	<tid>	RFC only: Open an SAP transaction and return a <tid> from SAP.
ConfirmTID	String/text	<tid>	RFC only: Confirms all RFC calls run in the current transaction <tid>.

Parameter	Data Type/ HTML Type/ Validator/ Size/MaxSize	Value/ Default	Description
IDocPathName	String/text		File-based IDoc RFC only: Specifies the name and path of the IDoc or status message from SAP system perspective.
IsStatusMessage	String/text/ Number	0,1	File-based IDoc RFC only: Indicates whether the transferred file is a status message. (0=IDoc, 1=status Message)
AutoCommit	String/text/ Number	0,1	BAPI only: Causes the SAP system to commit the BAPI call automatically after execution.
ServerResponse	String/text/ Number	0,1	The Server Response flag (ServerResponse = 1) indicates that a response document for a waiting RFC outbound should be returned to SAP. If the Server Response flag is set to 1, then the ServerSessionID and ServerSessionSequenceNumber must be passed to the SAP Suite instance to uniquely identify the waiting session with an RFC server instance.
ServerSessionID	String/text		The ServerSessionID uniquely identifies the destination RFC server instance of an SAP Suite instance for a server response. The ServerSessionID is generated by the RFC server and written into the process data of the started outbound business process.
ServerSessionSequence Number	String/text		The ServerSessionSequenceNumber uniquely identifies the session within the RFC server instance (identified by ServerSessionID) waiting for the response. The ServerSessionSequenceNumber is generated by the RFC server and written to the process data of the started outbound business process.

Business Process Definition Parameters

The following table describes the usage options of the business process parameters (IP = Instance Parameter, WP = Workflow Parameter):

Parameter	Type	Data Type/HTML Type/Validator/Size/MaxSize	Value/Default	sRFC	BAPI	RFC IDoc	ALE IDoc	Client or Server
CloseSession	WP	String/text/Number	0,1	O	O	-	-	C
KeepSessionOpen	WP	String/text/Number	0,1	O	O	-	-	C
SessionID	WP	String/text/	<sid>	O	O	-	-	C
BapiCommitWait	WP	String/text	0, 1	-	4.5: O 4.0: -	-	-	C
BapiCommit			0, 1		O			
BapiRollback			0, 1		O			
RFCModuleName	WP	String/text		M	M	-	-	C
Encoding	IP/WP	String/text	UTF-8	O	O	-	-	C
CreateTID	WP	String/text	<sid>	O	O	O	O	S
ConfirmTID	WP	String/text	<sid>	O	O	O	O	S
IDocPathName	WP/IP	String/text		-	-	M	-	S
IsStatusMessage	WP	String/text/Number	0,1	-	-	O	-	C
AutoCommit	WP	String/text/Number	0,1	-	O	?	O	C
Serverresponse	WP	String/text/Number	0,1	O	O			S
ServerSessionID	WP	String/text		O	O			S
ServerSessionSequenceNumber	WP	String/text		O	O			S

Export Parameters

The following table describes the business process parameters when exporting parameters:

Parameter	Data Type/HTML Type/Validator	sRFC	BAPI	RFC IDoc	ALE IDoc
SessionID	String	O	O	-	-

Parameter	Data Type/ HTML Type/Validator	sRFC	BAPI	RFC IDoc	ALE IDoc
TransactionID	String	O	O	-	-

TID Management for SAP Outbound

The SAP Suite adapter manages transactional integrity for SAP outbound to confirm that an IDoc packet has been successfully processed once. The SAP R/3 system assigns a transaction ID (TID) for every IDoc packet. The SAP Suite adapter stores the TID in the Application database in the SAP_TID table. Each row in the following table represents a separate IDoc packet and contains the following rows:

Row	Description
DATE_TIME	Date and time at which the TID table was updated.
TID (PK)	SAP transaction ID.
STATE	<ul style="list-style-type: none"> ◆ CREATED – Received the TID from SAP R/3 ◆ EXECUTED – Received the IDoc message with the TID and committed the transaction ◆ ROLLEDBACK – Rolled back the IDoc packet from further processing ◆ CONFIRMED – IDoc message has been queued for processing and the transaction has been committed
WFID	ID of the business process that has been started for processing the IDoc packet.
INSTANCE_NAME (PK)	Name of the service configuration of the SAP Suite adapter that received the IDoc packet.

It is important that entries in the TID table can only be modified by the service configuration of the SAP Suite adapter that created the entry. This includes the following guidelines:

Multiple RFC servers connected to the same Program ID can share the same TID management.

RFC servers from different configurations of the SAP Suite adapter (and different Program IDs) *cannot* share the same TID management.

The Outbound Flow Process

The outbound flow proceeds as follows:

1. SAP R/3 sends a TID to the service configuration of the SAP Suite adapter to which the RFC server is registered on the matching program ID.
2. The RFC server receives the TID and checks the TID table to determine whether it has previously received this TID from SAP R/3 or not. If the TID is not found in the TID table, then the SAP Suite adapter appends an entry to the TID file, specifying the date-time stamp, TID, and the state

(CREATED). The SAP Suite adapter returns a code to SAP R/3 indicating whether the TID was found, and the TID state determines whether SAP R/3 continues processing.

3. If SAP R/3 continues processing, the SAP Suite adapter starts a new transaction.
4. SAP R/3 sends the IDoc packet associated with the TID to the same service configuration of the SAP Suite adapter for this RFC server.
5. The SAP Suite adapter receives the IDoc packet and processes the data according to the way that is defined in the outbound business process, such as splitting IDoc packets or making routing decisions based on a map file.
6. After processing the data, the SAP Suite adapter returns success or an SAP exception (for example, if the target queue is full) to SAP R/3.
7. Based on the status returned from the SAP Suite adapter, SAP R/3 instructs the same service configuration to commit or roll back the transaction and performs one of the following actions:
 - ◆ The SAP Suite adapter commits the transaction and updates the date-time stamp and state (EXECUTED) in the TID file.
 - ◆ The SAP Suite adapter call rolls back the transaction and updates the date-time stamp and state (ROLLBACK) in the TID file.
 - ◆ If the transaction is successfully committed, the SAP Suite adapter updates the date-time stamp and state (CONFIRMED) in the TID file.

Advanced Status Returned by the SAP Suite Adapter

The following table includes the advanced status that may be returned by the SAP Suite adapter:

Status	Description
No Primary Document	No primary document was available when the SAP Suite adapter started running. The primary document is needed for BAPI, ALE, and RFC and contains the data to send to SAP R/3.
Cannot Commit and Rollback at the same time, please check your settings!	BAPI only. In the business process, both the BapiCommit and BapiRollback parameters are set to 1 at the same time, which is an invalid configuration.
Instance Parameter is missing	A service configuration of the SAP Suite adapter that is needed for this mode is missing.
Found no Session to close	BAPI only. While trying to actively close a session by using explicit business process parameters, the specified session in the Session ID parameter was not found. Check assignment of Session ID parameter.
No SessionID in Processdata found	The SAP Suite adapter started with a command that awaits the Session ID to be set in the business process.
Can not Create and Confirm a TID at the same time	RFC only. The SAP Suite adapter received instructions to create and confirm a TID (Transaction ID) at the same time. These instructions need to be separated into two different calls.

Status	Description
The TID value was empty when trying to confirm a TID!	FC only. The SAP Suite adapter received instructions to confirm a TID, but the Transaction ID Parameter was empty.
WrongSessionID	Connection session not found. Session ID may be expired.
OpenConnectionError	Error while establishing connection. The reason is explained in the report.
DisconnectError	Exception caught during disconnection.
NotConnectionPool	The RFC server is not initialized. The Connection pool will not be available.
OpenConnectionError	Error while establishing connection.
LoadNativeError	Cannot load native middleware library. Check library path and availability of native libraries.
JCOError	The library reported a fatal internal error. The native library may not be installed correctly.
NotMoreSession	Could not create another session (Session Limit reached).
ExitBecauseShutDown	No more sessions available; therefore, the SAP Suite adapter stops running.
RepositoryNotInited	The repository is not initialized.
GetTemplateError	Error when trying to retrieve a FunctionTemplate.
MetaDataNotAvailable	The metadata is not available in the repository.
CleanupError	Error during cleaning up JDBC connection.
RfcCallError	Error invoking the SAP RFC call.
ListenerError	Error in logging file listener.
SAPTableError	Error in analysis of SAP table.
RfcServerError	Error or Exception occurred in RFC server.
DisconnectError	Exception detected during disconnection.
PoolThreadError	Error in the connection pool test thread.
CreateTIDLogError	JDBC did not create any rows in Transaction Management table.
DeleteTIDLogError	JDBC did not delete a row in Transaction Management table.
LoadPropertyError	Error while loading properties from properties file.
ToDomConvertError	Error during the conversion to Document.
UpdateTIDLogError	JDBC did not update a row in Transaction Management table.
TidMaintenanceError	Error during maintenance of SAP_TID table.
OpenConnectionError	Error while establishing connection.

Status	Description
BapiDiscpatcherTransformer Exception	A problem occurred while mapping the Return Structure into process data.
BytesToDOMIOException	An IO Problem occurred while transferring Bytes into a DOM.
BytesToDOMParserException	The parser was not configured correctly while transferring bytes into a DOM.
BytesToDOMSaxException	A Sax problem occurred while transferring bytes into a DOM.
DomToBytesIOException	An IO problem occurred while transferring a DOM to bytes.
DomToBytesSaxException	A Sax problem occurred while transferring a DOM to bytes.
DomToStringIOException	An IO problem occurred while transferring a DOM to a string.
DomToStringSaxException	A Sax problem occurred while transferring a DOM to a string.
ParserConfigurationException	Parser configuration not property defined.
SENDALEAbapException	(SendALE) An Abap exception occurred.
SENDALEJcoException	(SendALE) A JCO exception occurred.

Usage Examples

The following sections contain examples using the SAP Suite adapter for inbound and outbound processing.

Inbound and Outbound IDoc Processing Preconditions

The inbound and outbound IDoc examples in the following sections focus only on the SAP Suite adapter configuration and the SAP inbound and outbound business processes for sending and receiving IDocs. To use the SAP inbound and outbound business processes to process EDI files, you need to perform the following additional steps:

1. Create maps that translate EDI to IDoc format (inbound processing) and IDoc format to EDI (outbound processing).
2. Create a business process for sending the translated IDoc to EDI file to the trading partner (outbound processing).
3. Define inbound and outbound EDI envelopes based on the EDI type (such as EDIFACT and X12) and:
 - ◆ Specify the translation maps for inbound and outbound processing.
 - ◆ Specify the SAP inbound business process for sending IDocs to SAP, and the business process for sending translated IDoc to EDI files to the trading partner.
4. Configure SAP inbound and outbound routes for the inbound and outbound IDocs. See *Configuring an Inbound Route* on page 1216 and *Configuring an Outbound Route* on page 1217.
5. Configure SAP inbound and outbound cross-references to tie the SAP inbound route and SAP outbound route to the appropriate trading partner EDI envelopes. See *Configuring an Inbound Cross-Reference* on page 1219 and *Configuring an Outbound Cross-reference* on page 1219.

Inbound Processing Examples

This section contains examples for the following inbound processing scenarios:

Sending an IDoc using Application Linking and Enabling (ALE) to an SAP system

Starting an SAP Business Application Programming Interface (BAPI) module to retrieve and return company information to Application

Sending an IDoc Using ALE

The following example illustrates an SAP Suite adapter configuration used for sending an IDoc document using ALE technology:

▶ ExampleGPMSAPSuiteALEInboundIDoc	
Service Settings	
Service Type	SAP Suite Adapter
Description	Example for documentation
System Name	ExampleGPMSAPSuiteALEInboundIDoc
Group	None
SAP Integration Mode	ALE based IDoc
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	None provided
Gateway Service	None provided
SAP System Number	99
SAP Version	4
Retry Sending IDocs	Yes
Max. Retries (0=unlimited)	5
Retry Sending Interval	10
Client	999
User	username
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	No
RFC Server Instances	1
Program ID	Ask_SAP_Administrator

Specifies the integration mode for interacting with the SAP system.

(Screen 1 of 2)

Outbound Process to start	None provided
Outbound Encoding	UTF8
Wait for synchronous RFC Outbound response	No
Response timeout (seconds, 0=unlimited)	0
Delete TIDs automatically	No
Delete TIDs after (days)	30
Register Remote Function Calls (BAPI or RFC Mode only)	None (for IDoc modes or BAPI/RFC Inbound)
Filter online RFC list by	None provided
Use hard max. connections limit (Off= soft limit)	No
Max. Connections	4
Soft Limit Delay Time (seconds)	120
Connection Check Interval (seconds)	30
Connection Idle Time (seconds)	240
Close session after maximum session time	Yes
Max. Session Time (minutes)	60
User	admin

(Screen 2 of 2)

The following example using the GPM illustrates a business process that uses the SAP Suite adapter to send an IDoc using ALE technology to an SAP system. The dimmed values were specified on the SAP Suite adapter configuration instance.

Service Editor - Send IDoc to SAP via ALE

Name: Send IDoc to SAP via ALE

Config: ExampleGPM SAP Suite ALE Inbound IDoc

Message To Service: Message From Service

Output Msg: Obtain Process Data first, then Messages

Message Name: SapSuiteInputMessage

Name	Value	Use XPAT
Ashost	127.0.0.1	<input type="checkbox"/>
AutoCommit		<input type="checkbox"/>
BapiCommit		<input type="checkbox"/>
BapiCommitWait		<input type="checkbox"/>
BapiRollback		<input type="checkbox"/>
BPName	[Not Applicable]	<input type="checkbox"/>
Client	999	<input type="checkbox"/>
CloseSession		<input type="checkbox"/>
Codepage	1100	<input type="checkbox"/>
ConfirmTID		<input type="checkbox"/>
ConnCheckInterval	30	<input type="checkbox"/>
ConnIdleTime	240	<input type="checkbox"/>
CreateTID		<input type="checkbox"/>

(Screen 1 of 4)

DeleteTIDAfterDays	30	<input type="checkbox"/>
Encoding		<input type="checkbox"/>
Group		<input type="checkbox"/>
Gwhost		<input type="checkbox"/>
Gwserv		<input type="checkbox"/>
HardLimit	No	<input type="checkbox"/>
IDocPathName		<input type="checkbox"/>
IsStatusMessage		<input type="checkbox"/>
JCOTraceLevel	0	<input type="checkbox"/>
JCOTracePath		<input type="checkbox"/>
KeepSessionOpen		<input type="checkbox"/>
Lang	EN	<input type="checkbox"/>
LoadBalanced	No	<input type="checkbox"/>
LocalDir		<input type="checkbox"/>
MapPath		<input type="checkbox"/>
MaxPoolSize	4	<input type="checkbox"/>
MaxRetry	5	<input type="checkbox"/>
MaxSessionTime	60	<input type="checkbox"/>

(Screen 2 of 4)

MaxStartupDelay	60		<input type="checkbox"/>
Mode	ALE based IDoc		<input type="checkbox"/>
Mshost			<input type="checkbox"/>
OutboundEncoding	UTF8		<input type="checkbox"/>
Passwd	*****		<input type="checkbox"/>
Port			<input type="checkbox"/>
ProgramID	Ask_SAP_Administrator		<input type="checkbox"/>
R3name			<input type="checkbox"/>
RetryInterval	10		<input type="checkbox"/>
RFCFilter	RFC		<input type="checkbox"/>
RFCList			<input type="checkbox"/>
RFCModuleName			<input type="checkbox"/>
RFCServerCount	1		<input type="checkbox"/>
RFCTrace	Off		<input type="checkbox"/>
RunServer	No		<input type="checkbox"/>
SAPVersion	4		<input type="checkbox"/>
SelectRFC	None (for IDoc modes or BAPIRFC Inbound)		<input type="checkbox"/>
SendRetry	Yes		<input type="checkbox"/>

Indicates that the IDoc is sent in ALE mode.

(Screen 3 of 4)

ServerResponse			<input type="checkbox"/>
ServerSessionID			<input type="checkbox"/>
ServerSessionSequenceNumber			<input type="checkbox"/>
SessionID			<input type="checkbox"/>
SessionMonitor	Yes		<input type="checkbox"/>
SoftLimitDelayTime	120		<input type="checkbox"/>
SyncWait	No		<input type="checkbox"/>
SyncWaitTimeout	0		<input type="checkbox"/>
Sysnr	99		<input type="checkbox"/>
TIDDeletion	No		<input type="checkbox"/>
TransactionID			<input type="checkbox"/>
User	username		<input type="checkbox"/>

(Screen 4 of 4)

The following example illustrates the same business process using BPML. The IDoc file orders.dat is the input passed to the business process and becomes the primary document. The primary document is input to the SAP Suite adapter.

```

<process name="ExampleSAPALEIDocInbound">
  <sequence name="Send IDoc">
    <operation name="Send IDoc to SAP via ALE">
      <participant name="ExampleGPMSSAPSuiteALEInboundIDoc"/>
      <output message="Xout">
        <assign to="." from="*" />
      </output>
      <input message="Xin">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>

```

SAP Suite adapter instance used to send the IDoc to SAP

The following example illustrates information returned from SAP to the business process, indicating that the SAP system transaction manager allowed Application to open a transaction:

```
<ProcessData>  
  <PrimaryDocument SCIOBJECTID="server11:754e616c:fb9c22589c:-70f" />  
  <TID>0A560DCB478E406B4D3502EF</TID>  
</ProcessData>
```

Transaction ID returned
from the SAP system and
placed in the process data
of the initiating business
process

Starting an SAP BAPI Module

The following example illustrates an SAP Suite adapter configuration used for starting a BAPI in an SAP system to retrieve company information.

► **ExampleGPM SAP Suite Inbound BAPI**

Service Settings

Service Type	SAP Suite Adapter
Description	Example for including in the documentation
System Name	ExampleGPM SAP Suite Inbound BAPI
Group	None
SAP Integration Mode	Synchronous BAPI
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	None provided
Gateway Service	None provided
SAP System Number	08
Client	999
User	uname
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	No
RFC Server Instances	1
Program ID	Ask_SAP_Administrator
Outbound Process to start	None provided
Outbound Encoding	UTF8

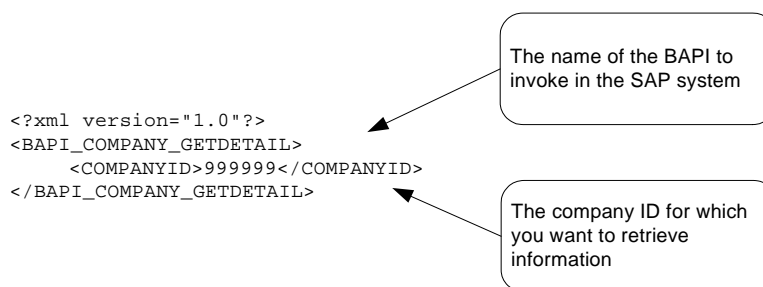
Specifies the integration mode for interacting with the SAP system.

(Screen 1 of 2)

Wait for synchronous RFC Outbound response	No
Response timeout (seconds, 0=unlimited)	0
Delete TIDs automatically	No
Delete TIDs after (days)	30
Register Remote Function Calls (BAPI or RFC Mode only)	None (for IDoc modes or BAPI/RFC Inbound)
Filter online RFC list by	None provided
Use hard max. connections limit (Off = soft limit)	No
Max. Connections	4
Soft Limit Delay Time (seconds)	120
Connection Check Interval (seconds)	30
Connection Idle Time (seconds)	240
Close session after maximum session time	Yes
Max. Session Time (minutes)	60
User	admin

(Screen 2 of 2)

The following example illustrates the input passed to the business process.



The following example using the GPM illustrates a business process that uses the SAP Suite adapter to start the BAPI_COMPANY_GETDETAIL BAPI. The dimmed values were specified in the SAP Suite adapter configuration instance.



Service Editor - CallCompanyGetDetailBusinessObject

Name: CallCompanyGetDetailBusinessObject

Config: ExampleGPM SAP Suite inbound BAPI

Message To Service: Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: SapSuiteInputMessage

Name	Value	Use XPATH?
Ashost	127.0.0.1	<input type="checkbox"/>
AutoCommit		<input type="checkbox"/>
BapiCommit		<input type="checkbox"/>
BapiCommitMail		<input type="checkbox"/>
BapiRollback		<input type="checkbox"/>
BPName	[Not Applicable]	<input type="checkbox"/>
Client	999	<input type="checkbox"/>
CloseSession		<input type="checkbox"/>
Codepage	1100	<input type="checkbox"/>

(Screen 1 of 5)

ConfirmTID		<input type="checkbox"/>
ConnCheckInterval	30	<input type="checkbox"/>
ConnIdleTime	240	<input type="checkbox"/>
CreateTID		<input type="checkbox"/>
DeleteTIDAfterDays	30	<input type="checkbox"/>
Encoding		<input type="checkbox"/>
Group		<input type="checkbox"/>
Gwhost		<input type="checkbox"/>
Gwserv		<input type="checkbox"/>
HardLimit	No	<input type="checkbox"/>
IDocPathName		<input type="checkbox"/>
IsStatusMessage		<input type="checkbox"/>
JCOTraceLevel	0	<input type="checkbox"/>
JCOTracePath	.	<input type="checkbox"/>
KeepSessionOpen		<input type="checkbox"/>
Lang	EN	<input type="checkbox"/>
LoadBalanced	No	<input type="checkbox"/>

(Screen 2 of 5)

LocalDir		<input type="checkbox"/>
MapPath		<input type="checkbox"/>
MaxPoolSize	4	<input type="checkbox"/>
MaxRetry	5	<input type="checkbox"/>
MaxSessionTime	60	<input type="checkbox"/>
MaxStartupDelay	60	<input type="checkbox"/>
Mode	Synchronous BAPI	<input type="checkbox"/>
Mshost		<input type="checkbox"/>
OutboundEncoding	UTF8	<input type="checkbox"/>
Passwd	*****	<input type="checkbox"/>
Port		<input type="checkbox"/>
ProgramID	Ask_SAP_Administrator	<input type="checkbox"/>
R3name		<input type="checkbox"/>
RetryInterval	10	<input type="checkbox"/>
RFCLiter	RFC	<input type="checkbox"/>
RFCList		<input type="checkbox"/>

Indicates synchronous BAPI mode.

(Screen 3 of 5)

RFCList			<input type="checkbox"/>
RFCModuleName	BAPI_COMPANY_GETDETAIL		<input checked="" type="checkbox"/>
RFCServerCount	1		<input type="checkbox"/>
RFCTrace	Off		<input type="checkbox"/>
RunServer	No		<input type="checkbox"/>
SAPVersion	4		<input type="checkbox"/>
SelectRFC	None (for IDoc modes or BAPIRFC Inbound)		<input type="checkbox"/>
SendRetry	Yes		<input type="checkbox"/>
ServerResponse			<input type="checkbox"/>
ServerSessionID			<input type="checkbox"/>
ServerSessionSequenceNumber			<input type="checkbox"/>
SessionID			<input type="checkbox"/>
SessionMonitor	Yes		<input type="checkbox"/>
SoftLimitDelayTime	120		<input type="checkbox"/>
SyncWait	No		<input type="checkbox"/>
SyncWaitTimeout	0		<input type="checkbox"/>

The name of the BAPI to invoke in the SAP system

(Screen 4 of 5)

Sysnr	08		<input type="checkbox"/>
TIDDeletion	No		<input type="checkbox"/>
TransactionID			<input type="checkbox"/>
User	uname		<input type="checkbox"/>

(Screen 5 of 5)

The following example illustrates the same business process using BPML.

```
<process name="ExampleCompanyGetDetail">
  <sequence name="Call To SAP Server">
    <operation name="CallCompanyGetDetailBusinessObject">
      <participant name="ExampleGPM SAPSuiteInboundBAPI"/>
      <output message="SAPOutput">
        <assign to="RFCModuleName" from="'BAPI_COMPANY_GETDETAIL'"/>
        <assign to="AutoCommit" from="0"/>
        <assign to="." from="*" />
      </output>
      <input message="SAPInput">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

The name of the BAPI to invoke in the SAP system.

The following example illustrates the company information returned from SAP to the business process as a primary document. The information is returned in XML format.

```
<BAPI_COMPANY_GETDETAIL>
  <COMPANYID>999999</COMPANYID>
  <COMPANY_DETAIL>
    <COMPANY>999999</COMPANY>
    <NAME1>HANDSOME, INC</NAME1>
    <NAME2 />
    <COUNTRY>USA</COUNTRY>
    <LANGU>E</LANGU>
    <STREET>5555 EAST MARTIN AVE</STREET>
    <PO_BOX />
    <POSTL_COD1>80220</POSTL_COD1>
    <CITY>DENVER</CITY>
```

```

<CURRENCY>DOLLAR</CURRENCY>
<COUNTRY_ISO>USA</COUNTRY_ISO>
<CURRENCY_ISO>DOLLAR</CURRENCY_ISO>
<LANGU_ISO>USA</LANGU_ISO>
</COMPANY_DETAIL>
<RETURN>
<TYPE/>
<CODE/>
<MESSAGE/>
<LOG_NO/>
<LOG_MSG_NO>000000</LOG_MSG_NO>
<MESSAGE_V1/>
<MESSAGE_V2/>
<MESSAGE_V3/>
<MESSAGE_V4/>
</RETURN>
</BAPI_COMPANY_GETDETAIL>

```

In addition, the SAP Suite returns session information from the SAP system and puts it in the process data of the initiating business process. For example:

```

<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <PrimaryDocument SCIObjID="server1:252e596c:fb4e22589c:-76b0" />
  <BapiCallReturnStructure>
    <RETURN>
      <TYPE/>
      <CODE/>
      <MESSAGE/>
      <LOG_NO/>
      <LOG_MSG_NO>000000</LOG_MSG_NO>
      <MESSAGE_V1/>
      <MESSAGE_V2/>
      <MESSAGE_V3/>
      <MESSAGE_V4/>
    </RETURN>
  </BapiCallReturnStructure>
</ProcessData>

```

Outbound Processing Examples

This section contains examples for the following outbound processing scenarios:

- Receiving a file-based IDoc from an SAP system using RFC
- Receiving an ALE/IDoc from an SAP system using RFC
- Receiving a request from SAP and returning a synchronous response using RFC

Receiving a File-based IDoc from SAP Using RFC

This section includes an example SAP Suite adapter configuration and the predefined SAPOutboundIDoc business process that runs when a file-based IDoc is received from an SAP system.

The following example illustrates an SAP Suite adapter configuration used for receiving a file-based IDoc from an SAP system.

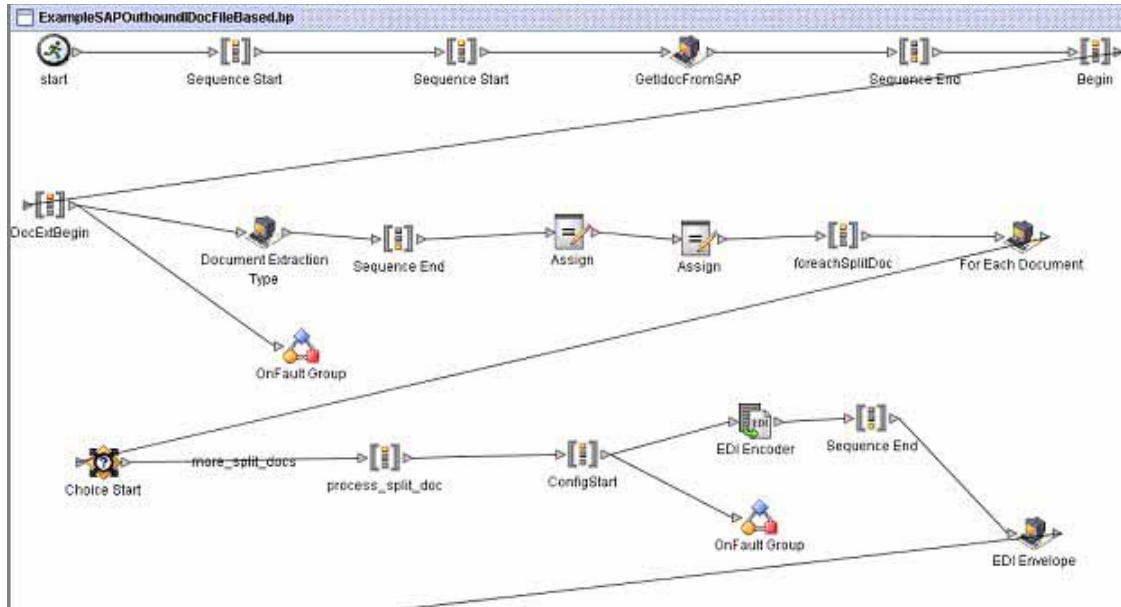
► **ExampleGPM SAP Outbound File Based IDoc**

Service Settings

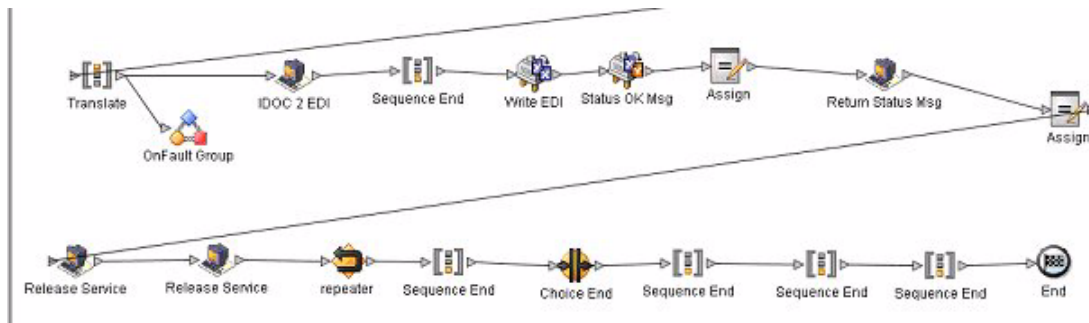
Service Type	SAP Suite Adapter	
Description	Configuration to include in documentation	
System Name	ExampleGPM SAP Outbound File Based IDoc	
Group	None	
SAP Integration Mode	File based IDoc RFC	Specifies the integration mode for interacting with the SAP system.
SAP system is loadbalanced	No	
SAP Application Server	127.0.0.1	
Gateway Host	127.0.0.1	
Gateway Service	sapgw01	
SAP System Number	01	
Client	999	
User	sapusername	
Password	*****	
Language	EN	
Code Page	1100	
Start RFC Server automatically	Yes	
RFC Server Instances	2	Business process invoked when a file-based IDoc is received from an SAP system
Program ID	SERVER_FRFC_CM	
Outbound Process to start	SAPOutboundIDoc	

The following example using the GPM illustrates the SAPOutboundIDoc business process that runs by the SAP Suite adapter for an outbound file-based IDoc. This business process retrieves the file-based IDoc from a directory on the SAP system and processes the file (translates the IDoc to EDI format and sends it to

a trading partner). In addition, the business process starts a subprocess that uses an instance of the SAP Suite adapter to send a status message back to the SAP system.



(Screen 1 of 2)



(Screen 2 of 2)

The following example illustrates the same business process using BPML.



```

<assign name="Assign" to="counter">1</assign>
<assign name="Assign" to="DOCUMENT_NAME_PREFIX">DOC-SPLIT-</assign>
<sequence name="foreachSplitDoc">
  <operation name="For Each Document">
    <participant name="ForEachDocument"/>
    <output message="ForEachDocumentTypeInputMessage">
      <assign to="DOCUMENT_NAME_PREFIX">DOC-SPLIT-</assign>
      <assign to="ITERATOR_NAME">DocExtract</assign>
      <assign to="." from="*" />
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
  <choice name="Choice Start">
    <select>
      <case ref="more_split_docs" activity="process_split_doc" />
    </select>
    <sequence name="process_split_doc">
      <sequence name="ConfigStart">
        <operation name="EDI Encoder">
          <participant name="EDIEncoder"/>
          <output message="EDIEncoderTypeInputMessage">
            <assign to="AcceptorLookupAlias" from="/*[name() =
concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/AcceptorLookupAlias/text()"/>
            <assign to="ReceiverID" from="/*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text())]/ReceiverID/text()"/>
            <assign to="SenderID" from="/*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text())]/SenderID/text()"/>
            <assign to="." from="*" />
          </output>
          <input message="inmsg">
            <assign to="." from="*" />
          </input>
        </operation>
      </sequence>
    </sequence>
  </choice>
</sequence>

```

Determines which envelope services need to run based on the AcceptorLookup Alias, SenderID, and ReceiverID.

Loops through each batch of IDocs and processes each document.

```

        <onFault>
            <sequence name="Sequence Start">
                <operation name="Status 04 Msg">
                    <participant name="Translation"/>
                    <output message="TranslationTypeInputMessage">
                        <assign to="." from="*"/>
                        <assign to="map_name">StatusMsg04</assign>
                        <assign to="PrimaryDocument" from="//
*[name() = concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/@SCIOBJECTID"/>
                    </output>
                    <input message="inmsg">
                        <assign to="." from="*"/>
                    </input>
                </operation>
                <assign name="Assign" to="IsStatusMessage">1</
assign>

                <operation name="Return Status Msg">
                    <participant name="InvokeSubProcessService"/>
                    <output
message="InvokeSubProcessServiceTypeInputMessage">
                        <assign to="INVOKE_MODE">ASYNC</assign>
                        <assign to="NOTIFY_PARENT_ON_ERROR">ALL</
assign>

                        <assign to="WFD_NAME">SAPinbDelivery</
assign>

                        <assign to="." from="*"/>
                    </output>
                    <input message="inmsg">
                        <assign to="." from="*"/>
                    </input>
                </operation>
                <assign name="Assign" to="counter" from="counter +
1"/>

                <repeat name="repeater" ref="foreachSplitDoc"/>
            </sequence>
        </onFault>
    </sequence>
    <operation name="EDI Envelope">
        <participant name="EDIEnvelope"/>
        <output message="EDIEnvelopeTypeInputMessage">
            <assign to="MODE">IMMEDIATE</assign>
            <assign to="RECEIVER_ID" from="ReceiverID/text()"/>
            <assign to="SENDER_ID" from="SenderID/text()"/>
            <assign to="." from="*"/>
        </output>
        <input message="inmsg">
            <assign to="." from="*"/>
        </input>
    </operation>

```

Determines which EDI enveloping business process needs to run to apply the EDI envelopes.

```

<sequence name="Translate">
  <operation name="IDOC 2 EDI">
    <participant name="InvokeSubProcessService"/>
    <output
message="InvokeSubProcessServiceTypeInputMessage">
      <assign to="INVOKE_MODE">SYNC</assign>
      <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
      <assign to="WFD_NAME">X12Envelope</assign>
      <assign to="." from="*" />
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
  <onFault>
    <sequence name="Sequence Start">
      <operation name="Status 05 Msg">
        <participant name="Translation"/>
        <output
message="TranslationTypeInputMessage">
          <assign to="." from="*" />
          <assign to="map_name">StatusMsg05</
assign>
          <assign to="PrimaryDocument"
from="//*[name() = concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/@SCIOBJECTID
"/>
          </output>
          <input message="inmsg">
            <assign to="." from="*" />
          </input>
        </operation>
        <assign name="Assign" to="IsStatusMessage">1</
assign>
        <operation name="Return Status Msg">
          <participant name="InvokeSubProcessService"/
>
          <output
message="InvokeSubProcessServiceTypeInputMessage">
            <assign to="." from="*" />
            <assign to="INVOKE_MODE">ASYN</assign>
            <assign
to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
            <assign to="WFD_NAME">SAPinbDelivery</
assign>
            </output>
            <input message="inmsg">
              <assign to="." from="*" />
            </input>
          </operation>
          <assign name="Assign" to="counter" from="counter
+ 1"/>
          <operation name="Release Service">
            <participant name="SAPRelease"/>
            <output
message="ReleaseServiceTypeInputMessage">
              <assign to="TARGET">/ProcessData/
*[local-name() = 'DOC']</assign>
              </output>
              <input message="inmsg">
                </input>
            </operation>
            <repeat name="repeater" ref="foreachSplitDoc"/>
          </sequence>
        </onFault>
      </sequence>
    </sequence>
  </sequence>

```

Invokes the appropriate EDI enveloping business process to load the translation map and perform IDoc to EDI translation. After translation is complete, it invokes the business process specified in the envelope to complete processing and send the EDI file to the trading partner.

Writes the translated EDI document to the local file system.

```
<operation name="Write EDI">  
  <participant name="FS_WriteEDI"/>  
  <output message="FileSystemInputMessage">  
    <assign to="assignedFilename" from="concat(//  
DOCUMENT_NAME_PREFIX/text(), //counter/text())"/>  
    <assign to="assignFilename">true</assign>  
    <assign  
to="extractionFolder">Enter_EDI_Extraction_Path</assign>  
    <assign to="." from="*" />  
  </output>  
  <input message="inmsg">  
    <assign to="." from="*" />  
  </input>  
</operation>
```

Constructs a status message to send back to the SAP system.

```
<operation name="Status OK Msg">  
  <participant name="Translation"/>  
  <output message="TranslationTypeInputMessage">  
    <assign to="." from="*" />  
    <assign to="map_name">StatusMsg</assign>  
    <assign to="PrimaryDocument" from="/*[name() =  
concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/@SCIObjectID  
" />  
  </output>  
  <input message="inmsg">  
    <assign to="." from="*" />  
  </input>  
</operation>  
<assign name="Assign" to="IsStatusMessage">1</assign>
```

```

<operation name="Return Status Msg">
  <participant name="InvokeSubProcessService"/>
  <output
message="InvokeSubProcessServiceTypeInputMessage">
    <assign to="INVOKE_MODE">ASYNC</assign>
    <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
    <assign to="WFD_NAME">SAPinbDelivery</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
<assign name="Assign" to="counter" from="counter + 1"/>
<operation name="Release Service">
  <participant name="SAPRelease"/>
  <output message="ReleaseServiceTypeInputMessage">
    <assign to="TARGET">/ProcessData/*[local-name() =
'BATCH']</assign>
    </output>
    <input message="inmsg">
    </input>
  </operation>
<operation name="Release Service">
  <participant name="SAPRelease"/>
  <output message="ReleaseServiceTypeInputMessage">
    <assign to="TARGET">/ProcessData/*[local-name() =
'GS']</assign>
    </output>
    <input message="inmsg">
    </input>
  </operation>
  <repeat name="repeater" ref="foreachSplitDoc"/>
</sequence>
</choice>
</sequence>
</sequence>
</sequence>
</process>

```

Business process invoked to send a status message to the SAP system. The status message is in IDoc format.

The following example illustrates the status message sent to the SAP system upon successful completion of the outbound business process that ran.

EDI_D340	90000000000000692334	24	3	Control information of EDI subsystem OK
EDI_D340	90000000000000692334	06	3	Translation OK
EDI_D340	90000000000000692334	08	3	Syntax check OK
EDI_D340	90000000000000692334	10	3	Interchange handling OK

An SAP administrator can then view the status messages in the SAP system.

Receiving an ALE IDoc from SAP Using RFC

This section includes an example SAP Suite adapter configuration and the predefined SAPOutboundALE business process that runs when an IDoc is received from an SAP system using ALE technology.

The following example illustrates an SAP Suite adapter configuration used for receiving an IDoc from an SAP system using ALE Technology.

► ExampleGPM SAPOutboundALE

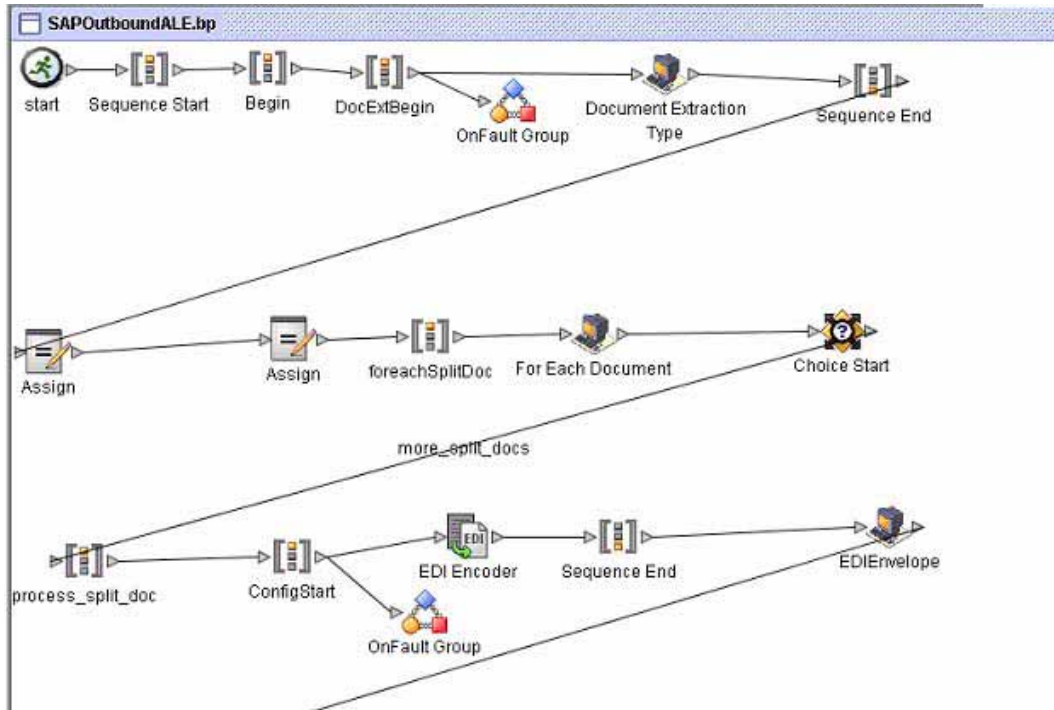
Service Settings	
Service Type	SAP Suite Adapter
Description	Configuration to include in documentation
System Name	ExampleGPM SAPOutboundALE
Group	None
SAP Integration Mode	ALE based IDoc
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	127.0.0.1
Gateway Service	sapgw01
SAP System Number	01
SAP Version	4
Retry Sending IDocs	Yes
Max. Retries (0=unlimited)	5
Retry Sending Interval	10
Client	999
User	sapusername
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	Yes
RFC Server Instances	1
Program ID	SERVER_ALE_TEST
Outbound Process to start	SAPOutboundALE

Specifies the integration mode for interacting with the SAP system.

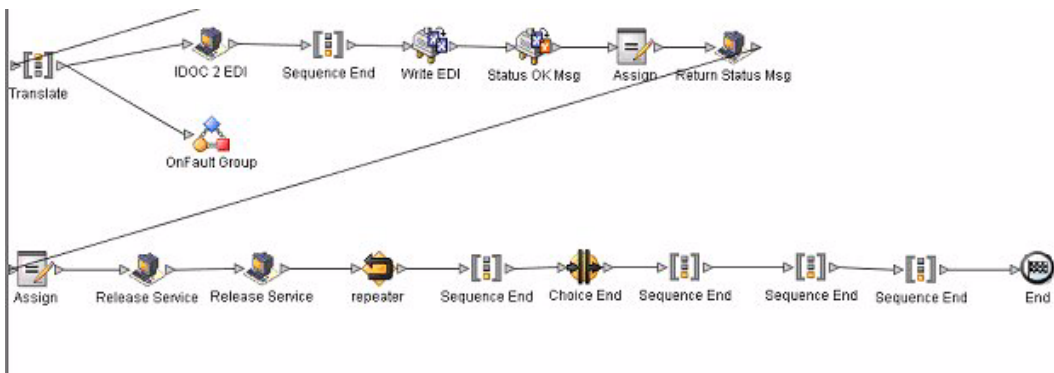
Business process invoked when an ALE IDoc is received from an SAP system

The following example using the GPM illustrates the SAPOutboundALE business process that runs by the SAP Suite adapter for an outbound ALE IDoc. With ALE technology, the IDoc is included in the outbound request and becomes the primary document for the outbound business process that ran. The business process processes the IDoc (translates the IDoc to EDI format and sends it to a trading partner). In addition, the

business process starts a subprocess that uses an instance of the SAP Suite adapter to send a status message back to the SAP system.



(Screen 1 of 2)



(Screen 2 of 2)

The following example illustrates the same business process using BPML.

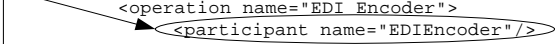


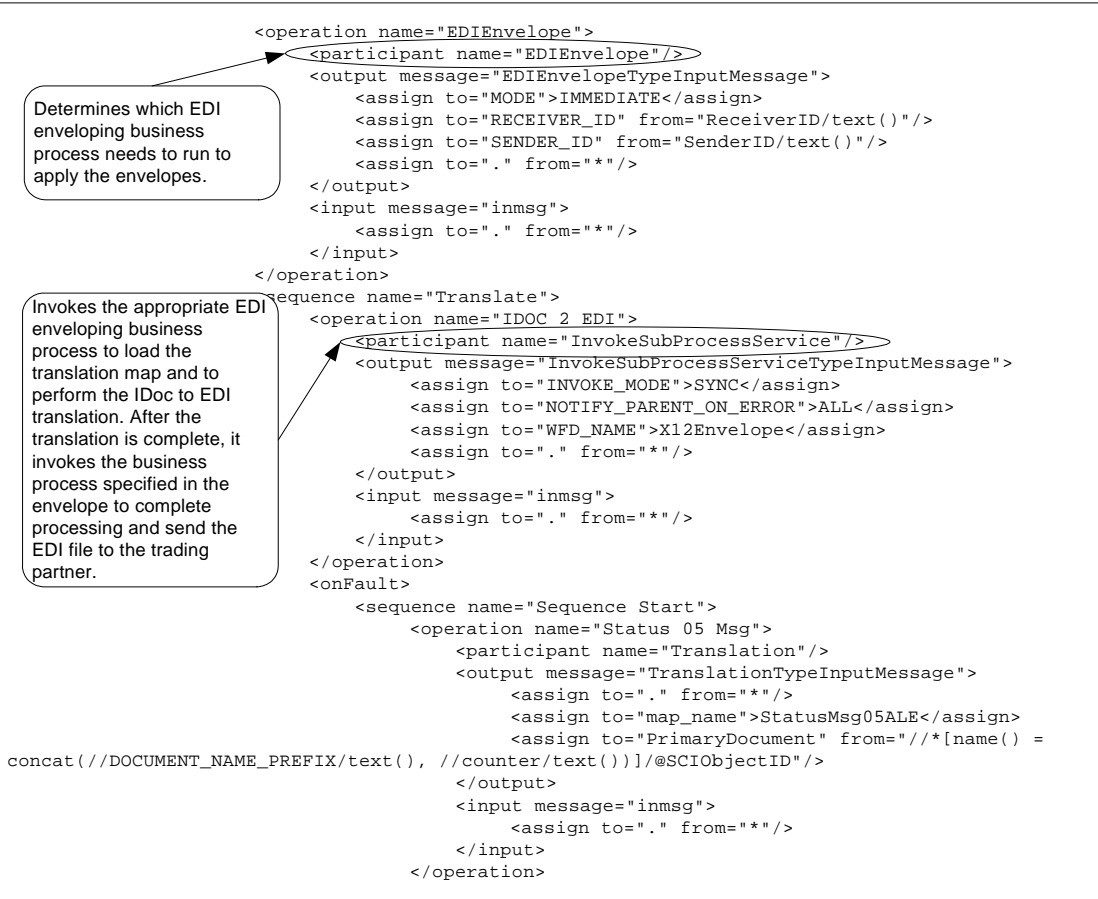
```

<choice name="Choice Start">
  <select>
    <case ref="more_split_docs" activity="process_split_doc"/>
  </select>
  <sequence name="process_split_doc">
    <sequence name="ConfigStart">
      <operation name="EDI Encoder">
        <participant name="EDI Encoder"/>
        <output message="EDIEncoderTypeInputMessage">
          <assign to="AcceptorLookupAlias" from="//*[name() =
concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())/AcceptorLookupAlias/text()"/>
          <assign to="ReceiverID" from="//*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text()/ReceiverID/text()"/>
          <assign to="SenderID" from="//*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text()/SenderID/text()"/>
          <assign to="." from="*/>
        </output>
        <input message="inmsg">
          <assign to="." from="*/>
        </input>
      </operation>
    </sequence>
    <onFault>
      <sequence name="Sequence Start">
        <operation name="Status 04 Msg">
          <participant name="Translation"/>
          <output message="TranslationTypeInputMessage">
            <assign to="." from="*/>
            <assign to="map_name">StatusMsg04ALE</
assign>
            <assign to="PrimaryDocument" from="//
*[name() = concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text()/@SCIOBJECTID"/>
            </output>
            <input message="inmsg">
              <assign to="." from="*/>
            </input>
          </operation>
          <assign name="Assign" to="IsStatusMessage">1</assign>
          <operation name="Return Status Msg">
            <participant name="InvokeSubProcessService"/>
            <output
message="InvokeSubProcessServiceTypeInputMessage">
              <assign to="INVOKE_MODE">ASYNC</assign>
              <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
              <assign to="WFD_NAME">SAPALEDelivery</assign>
              <assign to="." from="*/>
            </output>
            <input message="inmsg">
              <assign to="." from="*/>
            </input>
          </operation>
          <assign name="Assign" to="counter" from="counter + 1"/>
          <repeat name="repeater" ref="foreachSplitDoc"/>
        </sequence>
      </onFault>
    </sequence>
  </choice>

```

Determines which envelope services need to run based on the AcceptorLookupAlias, SenderID, and ReceiverID.



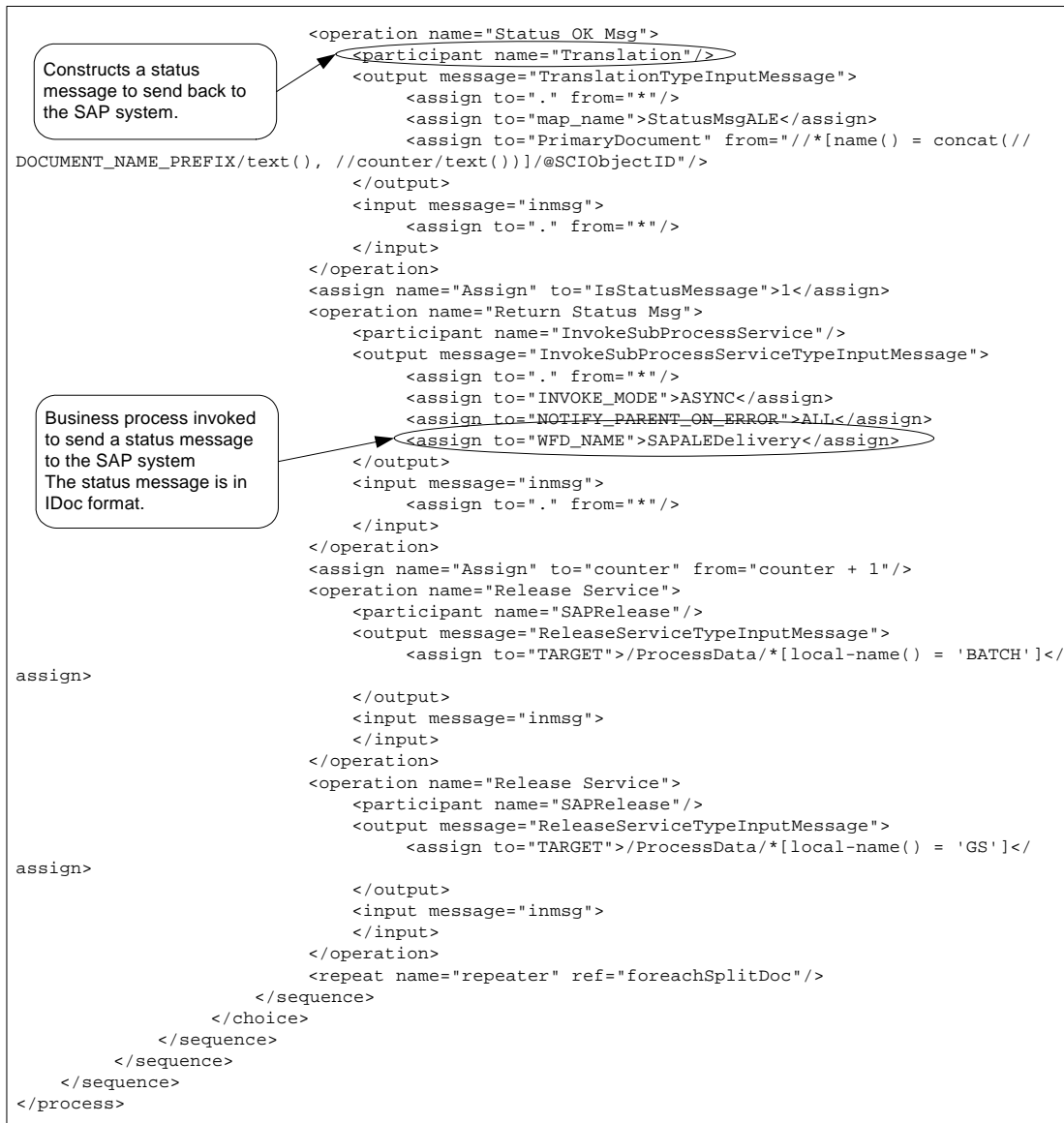


```

<assign name="Assign" to="IsStatusMessage">1</assign>
<operation name="Return Status Msg">
  <participant name="InvokeSubProcessService"/>
  <output message="InvokeSubProcessServiceTypeInputMessage">
    <assign to="." from="**"/>
    <assign to="INVOKE_MODE">ASYNC</assign>
    <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
    <assign to="WFD_NAME">SAPALEDelivery</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="**"/>
  </input>
</operation>
<assign name="Assign" to="counter" from="counter + 1"/>
<operation name="Release Service">
  <participant name="SAPRelease"/>
  <output message="ReleaseServiceTypeInputMessage">
    <assign to="TARGET"/>ProcessData/*[local-name() = 'DOC']</assign>
  </output>
  <input message="inmsg">
  </input>
</operation>
<repeat name="repeater" ref="foreachSplitDoc"/>
</sequence>
</onFault>
</sequence>
<operation name="Write EDI">
  <participant name="FS_WriteEDI"/>
  <output message="FileSystemInputMessage">
    <assign to="assignedFilename" from="concat(//DOCUMENT_NAME_PREFIX/text(), //
counter/text())"/>
    <assign to="assignFilename">true</assign>
    <assign to="extractionFolder">/server1/oracle/idocs/</assign>
    <assign to="." from="**"/>
  </output>
  <input message="inmsg">
    <assign to="." from="**"/>
  </input>
</operation>

```

Writes the translated EDI document to the local file system.



The following example illustrates the status message sent to the SAP system upon successful completion of the outbound business process that ran.

```

EDI_DC40 900 46C 2 SYSTAT01 STATUS ALETSTPORTLI 0000001111
SAPI02 LS SITEST 20040408041857

E2STATS001 900 000001000000 EDI_DS40
900000000000006922012004040804185724 Sterling SAPSuite Control
information of EDI subsystem OK
S

E2STATS001 900 000002000000 EDI_DS40
900000000000006922012004040804185706 Sterling SAPSuite

```

```

TranslationOK
S
E2STATS001      900      000003000000  EDI_DS40
900000000000006922012004040804185708      Sterling SAPSuite      Syntax
checkOK
S
E2STATS001      900      000004000000  EDI_DS40
900000000000006922012004040804185710      Sterling SAPSuite
Interchange handling OK
S

```

An SAP administrator can then view the status messages in the SAP system.

Receiving a Request from SAP and Returning a Synchronous Response Using RFC

This section includes an example SAP Suite adapter configuration and an example business process that runs when a request is received from an SAP system that requires a synchronous response. For example, a trading partner might need a price list for a particular order item before fulfilling the order.

Preconditions:

- ◆ The RFC must exist in the SAP system so the SAP Suite adapter can retrieve the RFC metadata description for it.
- ◆ The RFC must be registered in the RFC server of the SAP Suite adapter so the RFC server can listen for the selected RFC call.

You register an RFC in the SAP Suite adapter configuration instance that receives outbound RFC requests.

Examples:

The following example illustrates a simple custom RFC module Z_TRIGGERSI that starts by the SAP system. This RFC module has two import parameters – PARAM and VALUE – and one export parameter – RES.

```

FUNCTION Z_TRIGGERSI.
*"-----
*" "Locale Interface:
*"  IMPORTING
*"    VALUE(PARAM) TYPE  STRING OPTIONAL
*"    VALUE(VALUE) TYPE  STRING OPTIONAL
*"  EXPORTING
*"    VALUE(RES) TYPE  STRING
*"-----
write 'test'.

ENDFUNCTION.

```

The following example illustrates an SAP Suite adapter configuration used for receiving the RFC request.

► **ExampleGPM SAP Suite Outbound RFC Sync**

Service Settings	
Service Type	SAP Suite Adapter
Description	Documentation example
System Name	ExampleGPM SAP Suite Outbound RFC Sync
Group	None
SAP Integration Mode	Synchronous RFC
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	127.0.0.1
Gateway Service	sapgw01
SAP System Number	01
Client	999
User	sapusername
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	Yes
RFC Server Instances	1
Program ID	SYNCT
Outbound Process to start	ExampleSAP Outbound RFC Sync Response
Outbound Encoding	UTF8
Wait for synchronous RFC Outbound response	Yes

Specifies the integration mode for interacting with the SAP system.

Business process invoked when a request for the specified RFC is received from an SAP system

(Screen 1 of 2)

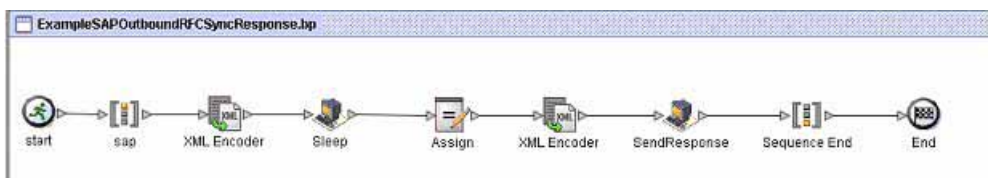
Response timeout (seconds, 0=unlimited)	120	
Delete TIDs automatically	No	
Delete TIDs after (days)	30	
Register Remote Function Calls (BAPI or RFC Mode only)	Enter RFCs offline	Option for registering RFCs
RFC List (comma separated)	Z_TRIGGERSI	The name of the RFC module to invoke
Filter online RFC list by	None provided	
Use hard max. connections limit (Off= soft limit)	Yes	
Max. Connections	4	
Soft Limit Delay Time (seconds)	120	
Connection Check Interval (seconds)	30	
Connection Idle Time (seconds)	240	
Close session after maximum session time	Yes	
Max. Session Time (minutes)	60	
User	admin	

(Screen 2 of 2)

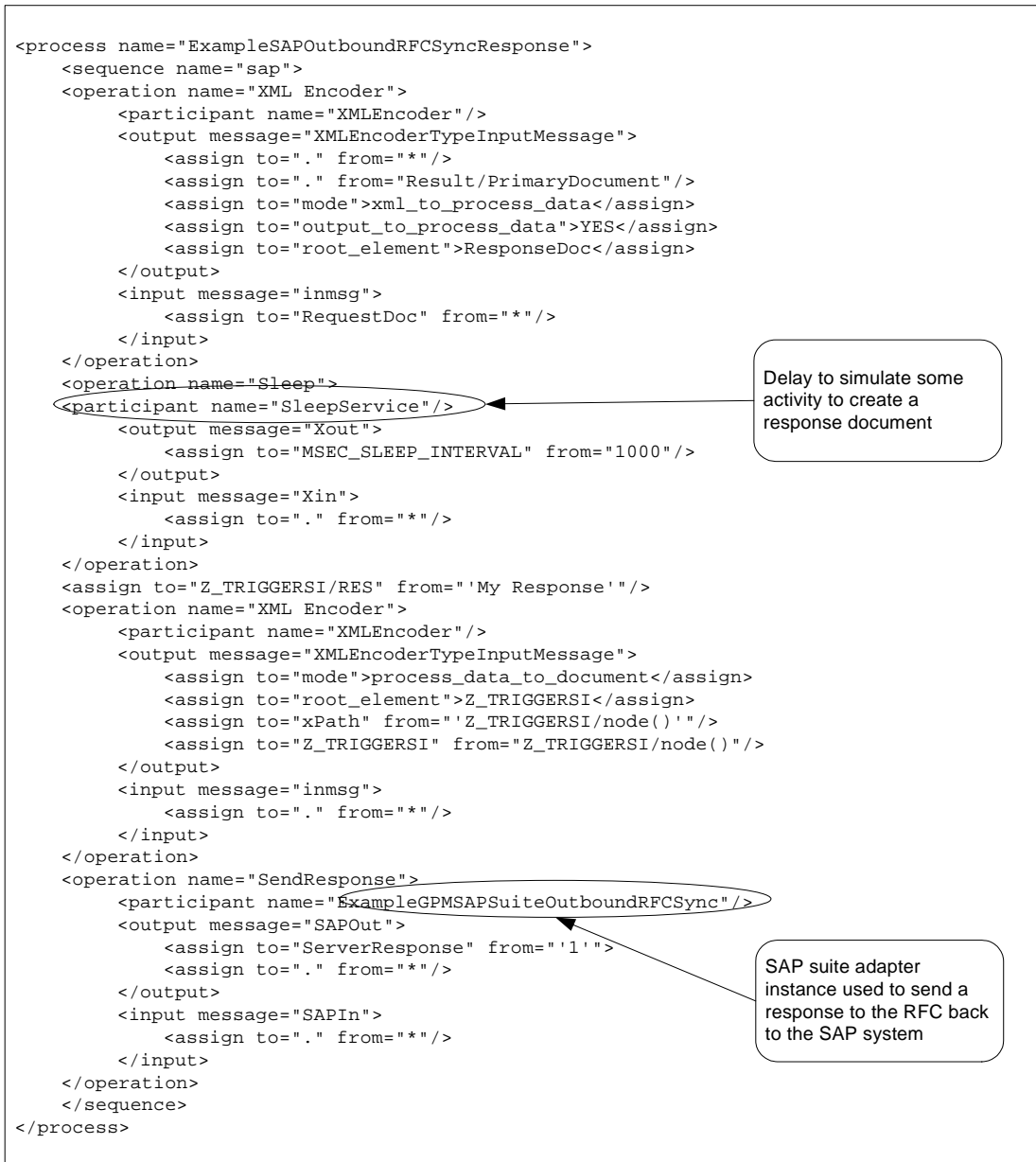
When an outbound RFC is detected by the SAP Suite adapter, the RFC server runs the business process specified on the SAP Suite adapter configuration. The RFC parameters are input to the business process and become the primary document. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Z_TRIGGERSI>
<PARAM>AAA</PARAM>
<VALUE>BBB</VALUE>
<RES></RES>
</Z_TRIGGERSI>
```

The following example using the GPM illustrates an example business process that starts by the SAP Suite adapter for an outbound RFC request. This business process creates and returns a response to the RFC back to SAP.



The following example illustrates the same business process using BPML.



Note: You must set the ServerResponse parameter to 1 and pass it to the SAP Suite adapter used for sending a response back to SAP. Additionally, you must pass the ServerSessionID and ServerSessionSequenceNumber parameters to the SAP Suite adapter. Both ServerSessionID and ServerSessionSequenceNumber parameters are passed using process data in the previous example. See *Business Process Definition Parameters* on page 1228 for a description of these parameters.

The following example illustrates the response sent back to the SAP system:

```

<Z_TRIGGERSI>
<RES>My Response</RES>

```

</Z_TRIGGERSI>

Connection Retry

When the SAP Suite adapter is started during the application startup, the RFC Server tries to establish a connection to the SAP System configured in the adapter instance. The default behavior of the RFC Server component is to keep trying to establish a connection until it successfully opens a connection.

However, for some permanent login error types, no retry is performed. For example, if the configuration has a wrong SAP user password, then no retry is performed because the error has to be resolved manually (by entering the correct password). Also, no retry is performed if a login fails because of a locked SAP user, which requires the SAP Administrator to manually unlock the user.

Sometimes, the SAP user that is used to login by the SAP Suite adapter is locked during SAP maintenance, and then unlocked after the maintenance. In this case, the *User locked* error must be treated as a temporary error condition and a connection retry is required so that the RFC Server can reconnect automatically after the SAP user is unlocked.

Perform the following steps to enable the connection retry in case of a locked SAP user for a SAP Suite adapter instance:

1. In the *install_dir/properties* directory, add the following line at the end of the *sap.properties* file:

```
SAP.instance_name UnconditionalConnectionRetry = Yes
```

instance_name must be replaced by the name of the SAP Suite adapter instance.

2. Save the file.
3. Restart the application.

Troubleshooting Tips

This section contains troubleshooting tips for using the SAP Suite adapter.

Java Error in the SAP Outbound Business Process

For the *SAPOutboundIDoc.bp*, if the *FS_WriteEDI* service (which is an instance of the File System adapter) is not configured properly to extract data, the Advanced Status column in the Business Process Monitor page displays the following Java error message:

```
java.io.FileNotFoundException
```

In addition, the Status Report column does not provide a report. In this circumstance, the *FS_WriteEDI* service is working as designed. However, the Advance Status Details will show that a file was not found occurs if the *FS_WriteEDI* service is not configured correctly.

SAP Suite Adapter (Build 4312 or higher)

The SAP Suite adapter for SAP R/3 enables you to integrate SAP R/3 versions 3.1i and later with non-SAP applications from vendors, such as PeopleSoft and Oracle, to conduct e-business with trading partners.

The SAP-certified (CA-ALE) SAP Suite adapter enables Application to support platform, data, process, and B2B integration for SAP R/3 by offering the following features:

- SAP inbound and outbound communications processing (ALE (Application Linking and Enabling) and IDOC (Intermediate Documents))

- Synchronous Remote Function Calls (RFCs) and responses

- Synchronous Business Application Programming Interface (BAPI) calls, including transaction handling

- Load balancing for SAP R/3 clusters

- Connection pooling and authentication

The following table provides an overview of the SAP Suite adapter:

System name	SAPSuite
Graphical Process Modeler (GPM) categories	All Services, Applications > ERP
Description	Provides a set of SAP R/3 integration methods.
Business usage	<ul style="list-style-type: none">◆ Near real-time and message-based integration of business processes between non-SAP back-end applications and SAP R/3 systems◆ Support of SAP interfaces: BAPI, synchronous RFC, file-based IDoc RFC (CA-EDI), and ALE/tRfc message-based IDoc (CA-ALE)
Usage example	BAPI example: <ol style="list-style-type: none">1 To determine availability of a certain material in a vendor's stock, you can use a synchronous BAPI call in SAP R/3 and receive the response immediately.2 You can then send a purchase order to buy the material.
Preconfigured?	Partially. You must create a configuration of the SAP Suite adapter and perform minimal configuration to the SAP business processes. The business processes enable SAP inbound and outbound IDoc processing.
Requires third party files?	Yes, SAP Java Connector (JCo) Libraries (most recent version within 2.1.x)
Platform availability	All supported Application platforms
Related services	None

Application requirements	SAP connection details (see <i>Before You Begin</i> on page 1278) are required for the SAP Suite adapter to communicate with SAP.
Initiates business processes?	The SAP Suite adapter initiates a business process in Application for the following instances: <ul style="list-style-type: none"> ◆ RFC outbound call ◆ Receipt of an IDoc from SAP R/3 in file-based IDoc RFC mode ◆ Receipt of an IDoc from SAP R/3 in ALE-based IDoc RFC mode
Invocation	The SAP Suite adapter can trigger an outbound business process by configuring the name of the business process. In addition, a business process can start the SAP Suite adapter to perform SAP inbound processing. Data received starts a predetermined business process on an SAP outbound RFC call. For inbound business processes to Application, it is event-driven.
Business process context considerations	For BAPI and synchronous RFC, the SAP Suite adapter creates a primary document in XML format and returns Session or Transaction IDs in the process data. For SAP IDoc outbound, the RFC server creates XML parameters in process data that describe relevant parameters for further processing.
Returned status values	<ul style="list-style-type: none"> ◆ Success ◆ Error – Review advanced status and process data content. Also see <i>Advanced Status Returned by the SAP Suite Adapter</i> on page 1303.
Restrictions	The only supported versions are: <ul style="list-style-type: none"> ◆ SAP R/3 versions 3.1i and later ◆ IDoc versions 3 and 4
Persistence level	None. The persistence is determined by the persistence level of the business processes used for processing inbound and outbound requests.
Testing considerations	Parameter in the Debug Log level and the RFC/JCo trace to get additional debug information on the SAP Suite adapter from library and RFC levels.

How the SAP Suite Adapter Works

To enable a connection to SAP from Application, SAP provides SAP Java™ Connector® (SAP JCo), a free software product for SAP customers which supports connections for the following platforms:

IBM® AIX®

HP-UX11

Linux®

Sun™ Solaris™

Microsoft® Windows®

IBM iSeries

Note: For many platforms, SAP offers 32-bit and 64-bit versions of the JCo libraries, depending on the operating system and the JVM (Java Virtual Machine). Make sure that you install the correct version.

Communicating and Processing IDocs

After installing SAP JCo and the SAP Suite adapter, you configure the adapter and use it in business processes. To communicate with SAP R/3 and your trading partners, and to process IDocs, Application provides several business processes that work together. These business processes use BPML activities, services, and adapters to retrieve documents, perform EDI and IDoc translation, and send documents. These business processes *must* be used together. To implement the business processes, you must perform minimal configuration.

There are two types of IDoc processing: file-based and ALE-based.

For file-based IDoc processing, Application provides the following business processes:

- SAP Inbound IDoc

- SAP Outbound IDoc

- SAP Inbound Delivery

For ALE/IDoc processing, Application provides the following business processes:

- SAP Outbound ALE

- SAP Delivery ALE

SAP Inbound IDoc (SAPInboundIDoc.bp)

SAPInboundIDoc runs after the translation of inbound EDI data to IDoc. Application envelope definitions are associated with SAP routes using the SAP cross-reference configuration. SAP Inbound IDoc inserts the proper routing information into the IDoc control record and transfers the completed IDoc to SAP using the SAP Inbound Delivery business process.

SAP Outbound IDoc (SAPOutboundIDoc.bp)

The SAP Suite adapter retrieves IDocs from SAP R/3. After retrieving IDocs, the SAP Outbound IDoc business process provides end-to-end processing of IDocs. The SAP Outbound IDoc business process enables IDocs to be grouped based on the user-provided EDI envelope definitions. While processing IDocs and preparing them for translation, the SAP Outbound IDoc business process generates status messages that describe processing results. After translation, the SAP Outbound IDocs business process calls the SAP Inbound Delivery business process, which uses FTP to send the status messages back to SAP R/3.

SAP Inbound Delivery (SAPinbDelivery.bp)

Both the SAP Inbound IDoc and SAP Outbound IDoc business processes use the SAP Inbound Delivery business process to send documents. That is:

- After the SAP Inbound IDoc business process performs the final translation on the IDoc (such as adding the routing information to the IDoc control record), the SAP Inbound Delivery business process uses FTP to send the IDocs to SAP R/3.

After the SAP Outbound IDoc business process translates an IDoc received from SAP R/3 to EDI, the SAP Inbound Delivery business process uses FTP to send status messages to SAP R/3.

SAP Outbound ALE (SAPOutboundALE.bp)

The SAP Suite adapter receives IDocs from SAP R/3 using ALE technology. After receiving IDocs, the SAP Outbound ALE business process provides end-to-end processing of IDocs. The SAP Outbound ALE business process enables IDocs to be grouped based on the user-provided EDI envelope definitions. While processing IDocs and preparing them for translation, the SAP Outbound ALE business process generates status messages that describe processing results. After translation, the SAP Outbound ALE business process calls the SAP Delivery ALE business process to send the status messages back to SAP R/3.

SAP Delivery ALE (SAPALEDelivery.bp)

The SAP Outbound ALE business process uses the SAP Delivery ALE business process to send status messages back to SAP R/3 after IDoc to EDI translation is complete.

For more information, see *Implementing the SAP R/3 Business Processes* on page 1287.

Business Scenario

Your company receives a purchase order from a trading partner in EDI format. You need to translate the EDI file to IDoc format and send it using file-based RFC to your back-end SAP system for further processing.

Business Solution Example

The approaches used to solve this business scenario includes the following tasks:

Configure EDI inbound envelopes for the purchase order. You need to define the following information in the appropriate envelopes:

- ◆ The map used to translate the EDI file to IDoc format
- ◆ The predefined SAPInboundIDoc business process as the business process to run after the EDI to IDoc translation is complete

Configure the SAP inbound route that contains the SAP ports and SAP client number of the SAP system that will receive the IDoc. See *Configuring an Inbound Route* on page 1289.

Configure the SAP inbound route cross-reference entry to link the SAP inbound route and EDI inbound envelope for routing the IDoc to the appropriate SAP system. See *Configuring an Inbound Cross-Reference* on page 1292.

Create a business process that includes the EDI Deenvelope service for deenveloping the EDI document.

Configure an instance of the SAP Suite adapter that uses file-based IDoc RFC (see *Example SAP Suite Adapter Configuration* on page 1267).

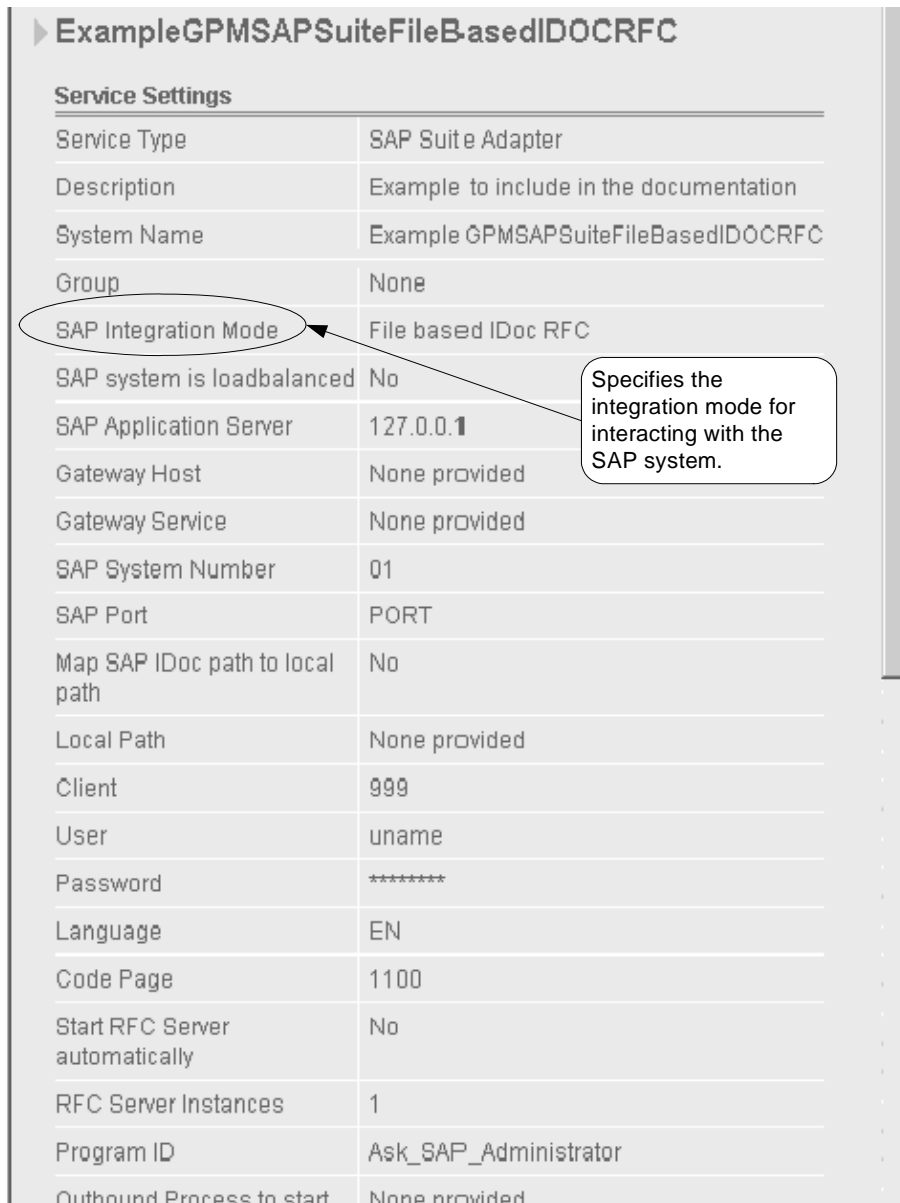
Update the predefined SAPInbDelivery business process with the appropriate values for sending the IDoc to the SAP system. See *SAP Inbound Delivery (SAPInbDelivery.bp)* on page 1265 for the required updates.

Create an FTP login authentication file required by the FTP server and place the file in the directory where Application is installed. See *Before You Begin* on page 1278.

This business solution example focuses only on the SAP Suite adapter configuration and the SAP Inbound IDoc business process.

Example SAP Suite Adapter Configuration

A sample configuration might look like the following:



Service Settings	
Service Type	SAP Suite Adapter
Description	Example to include in the documentation
System Name	ExampleGPM SAP Suite File Based IDOC RFC
Group	None
SAP Integration Mode	File based IDoc RFC
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	None provided
Gateway Service	None provided
SAP System Number	01
SAP Port	PORT
Map SAP IDoc path to local path	No
Local Path	None provided
Client	999
User	uname
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	No
RFC Server Instances	1
Program ID	Ask_SAP_Administrator
Outbound Process to start	None provided

(Screen 1 of 2)

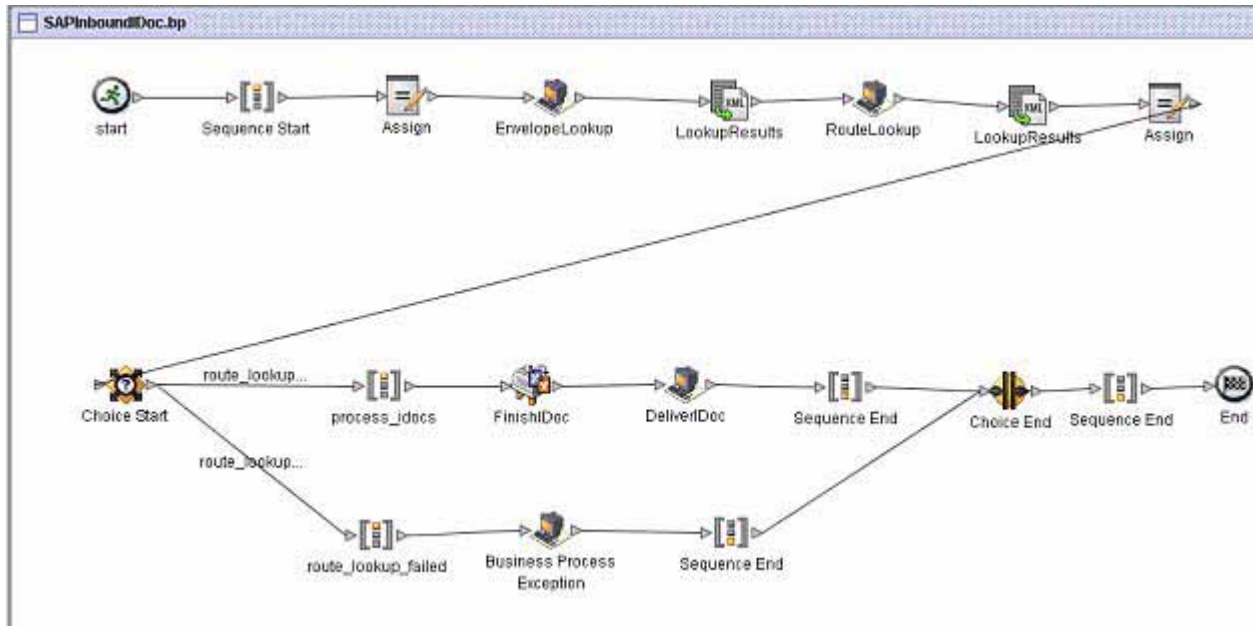
Outbound Process to start	None provided
Outbound Encoding	UTF8
Wait for synchronous RFC Outbound response	No
Response timeout (seconds, 0=unlimited)	0
Delete TIDs automatically	No
Delete TIDs after (days)	30
Register Remote Function Calls (BAPI or RFC Mode only)	None (for IDoc modes or BAPI/RFC Inbound)
Filter online RFC list by	None provided
Use hard max. connections limit (Off= soft limit)	No
Max. Connections	4
Soft Limit Delay Time (seconds)	120
Connection Check Interval (seconds)	30
Connection Idle Time (seconds)	240
Close session after maximum session time	Yes
Max. Session Time (minutes)	60
User	admin

(Screen 2 of 2)

Business Solution Example Business Processes

The following example shows the predefined SAPInboundIDoc business process in the GPM. This business process is specified on the EDI inbound envelope and runs after EDI Deenveloping and after the EDI to

IDoc translation is complete. The translated IDoc is input to the business process and becomes the primary document.



The following example shows the corresponding business process solution using BPML.

```

<process name="SAPInboundIDoc">
  <rule name="route_lookup_results">
    <condition>SAPXRef/IDOCControlData/SAPPORTNAME != &quot;&quot;</condition>
  </rule>
  <sequence name="Sequence Start">
    <assign name="Assign" to="OrigPrimaryDoc" from="PrimaryDocument/@SCIOBJECTID"/>
    <operation name="EnvelopeLookup">
      <participant name="RNJDBCAdapter"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="EnvelopeID/text()"/>
        <assign to="paramtype1">String</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="result_name">SAPXRefEnv</assign>
        <assign to="row_name">EnvelopeName</assign>
        <assign to="sql">select DISTINCT(NAME) from ENVELOPE where ENVELOPE_ID =
?</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="LookupResults">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">SAPXRefEnv</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="RouteLookup">
      <participant name="RNJDBCAdapter"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="SAPXRefEnv/EnvelopeName/NAME/text()"/>
        <assign to="paramtype1">String</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="result_name">SAPXRef</assign>
        <assign to="row_name">IDOCControlData</assign>
        <assign to="sql">select SAPPORTNAME, EDIPORTNAME, CLIENTNUM from
SAP_ROUTE, SAP_XREF where SAP_XREF.ENVELOPE = ? AND SAP_ROUTE.NAME = SAP_XREF.ROUTE</
assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="LookupResults">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">SAPXRef</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>

```

Queries the database for the name of the EDI envelope to retrieve the corresponding SAP route.

Queries the database for the SAP route information based on the EDI envelope name.

```

<assign name="Assign" to="PrimaryDocument" from="OrigPrimaryDoc/@SCIOBJECTID" />
<choice name="Choice Start">
  <select>
    <case ref="route_lookup_results" activity="process_idocs" />
    <case ref="route_lookup_results" negative="true"
activity="route_lookup_failed" />
  </select>
  <sequence name="process_idocs">
    <operation name="FinishIDoc">
      <participant name="Translation" />
      <output message="TranslationTypeInputMessage">
        <assign to="map_name">finishIDoc</assign>
        <assign to="output_to_process_data">NO</assign>
        <assign to="validate_input">NO</assign>
        <assign to="validate_input_against_dtd">NO</assign>
        <assign to="validate_output">NO</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <operation name="DeliverIDoc">
      <participant name="InvokeSubProcessService" />
      <output message="InvokeSubProcessServiceTypeInputMessage">
        <assign to="INVOKE_MODE">ASYNC</assign>
        <assign to="WFD_NAME">SAPinbDelivery</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
  <sequence name="route_lookup_failed">
    <operation name="Business Process Exception">
      <participant name="BPExceptionService" />
      <output message="BPExceptionServiceTypeInputMessage">
        <assign to="statusReport">Route Lookup Failed</assign>
        <assign to="." from="*" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</choice>
</sequence>
</process>

```

Adds the SAP routing information to the IDoc control record to route the IDoc to the correct SAP system.

Invokes this predefined business process for sending the IDoc to the SAP system.

The following example shows the predefined SAPinbDelivery business process used to send the IDoc to the SAP system.

```

<process name="SAPinbDelivery">
  <sequence name="sap">
    <operation>
      <participant name="GetDocumentInfoService"/>
      <output message="Xout">
        <assign to="." from="*" />
      </output>
      <input message="Xin">
        <assign to="IDocDocument" from="DocumentName" />
      </input>
    </operation>
    <operation name="FTP_Send">
      <participant name="FTPSend"/>
      <output message="FTP_SEND_ADAPTERInputMessage">
        <assign to="." from="*" />
        <assign to="xport-ftp-conntype">PASSIVE</assign>
        <assign to="xport-ftp-dir">/sapmnt/I02/global/nbtest</assign>
        <assign to="xport-ftp-document" from="/ProcessData/IDocDocument/
DocumentName/text()" />
        <assign to="xport-ftp-host">00.000.00.000</assign>
        <assign to="xport-ftp-mode">PUT</assign>
        <assign to="xport-ftp-port">21</assign>
        <assign to="xport-tp-authfile">/home/rhandsom/authfile</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <assign to="idocName" from="concat('/sapmnt/I02/global/nbtest/',/ProcessData/
IDocDocument/DocumentName/text())" />
    <operation name="TriggerSAP">
      <participant name="ExampleSAPSuiteFileBasedIDOCRFC" />
      <output message="SAPOut">
        <assign to="." from="*" />
        <assign to="IDocPathName" from="idocName/text()" />
      </output>
      <input message="SAPIn">
        <assign to="." from="*" />
      </input>
    </operation>
    <onFault>
      <assign name="Assign" to="PrimaryDocument">Sap Delivery Failed</assign>
    </onFault>
  </sequence>
</process>

```

The following example shows sample output returned to the business process when the IDoc has been successfully received by the SAP system.

```

<EDI_DATA_INCOMING>
<PATHNAME>/sapmnt/I02/directory/orders.dat</PATHNAME>
<PORT>EDI PORT</PORT>
</EDI_DATA_INCOMING>

```

The EDI_DATA_INCOMING element represents the SAP Remote Function Call (RFC) that started. The PATHNAME and PORT elements are the input parameters that were passed into the SAP RFC.

A Basic Status of “Success” and Advanced Status of “None” in the Business Process Modeler indicate that the SAP Suite adapter successfully executed the RFC and that the SAP system has verified the EDI_DC header segment of the IDoc.

Running the Business Solution Example

Assuming that you have completed all the prerequisite requirements for EDI to IDoc document exchanges, you need to perform the following steps to run the business solution example:

1. Run a business process that includes the EDI Envelope service and pass in the EDI file.
2. Verify the results in the Business Process Monitor.

See *Usage Examples* on page 1313 for additional examples of using the SAP Suite adapter.

Implementing the SAP Suite Adapter

To implement the SAP Suite adapter, complete the following tasks:

1. Activate your license for the SAP Suite adapter. For information, see *An Overview of Implementing Services*.
2. Install the SAP Java Connector. For information, see *Installing the SAP Java Connector* on page 1274.
3. Create all of the maps that will be used to translate the IDocs that you are enveloping, and check them into Application.
4. Set up your trading partner information, including EDI interchange, group, and transaction envelopes.
5. Create an SAP Suite adapter configuration. For information, see *Creating an SAP Suite Adapter Configuration* on page 1278.
6. Create business processes or configure/use the following predefined business processes to communicate and implement document processing:

For file-based IDoc processing:

- ◆ SAPInboundIDoc.bp
- ◆ SAPOutboundIDoc.bp
- ◆ SAPInbDelivery.bp

For ALE IDoc processing:

- ◆ SAPOutboundALE.bp
- ◆ SAPALEDelivery.bp

Note: Before you can use your business process, you must check the predefined business processes out of Application. After implementing the predefined business processes, you must check in your business process back to Application.

For more information about implementing SAP business processes, see *Implementing the SAP R/3 Business Processes* on page 1287.

7. Create inbound and outbound routes that describe the key fields in the IDoc that enable Application to route the IDoc to the appropriate trading partner. For information, see *Configuring an SAP R/3 Cross-Reference* on page 1292.
8. Create inbound and outbound cross-references to the routes that enable Application to tie EDI envelopes to their corresponding SAP routes (and vice versa) for routing documents to and from the external trading partners. For information, see *Configuring an SAP R/3 Cross-Reference* on page 1292.

Installing the SAP Java Connector

You must download and install the most recent version within 2.1.x of the SAP Java Connector (SAP JCo) before installing the SAP Suite adapter. The SAP JCo contains libraries and packages that support various platforms and enable the SAP Suite adapter to communicate with an SAP system and with Application. After obtaining the SAP JCo, you must make the files available to the host system where Application is installed.

Downloading the SAP Java Connector

To download the SAP Java Connector:

1. Create a user account to access the SAP Service Marketplace offerings.
2. On the same computer where you have installed Application and activated a license for the SAP Suite adapter, download the free SAP JCo from the SAP Service Marketplace Web site at <https://websmp101.sap-ag.de/>.
 - a. Log in to SAP Service Marketplace and access the SAP JCo download software from <http://service.sap.com/connectors>. If necessary, select the **Tools & Services** page to display the download page.
 - b. Download the most recent version within 2.1.x of the SAP JCo for your specific platform. The following table lists the specific files for each platform (included in the .tgz file on UNIX or the .zip file on Windows) that the SAP Suite adapter uses in Application:

Note: When transferring files from Windows to UNIX, make sure that you transfer the files in binary mode.

Platform	SAP JAVA Connector Files
IBM AIX	<ul style="list-style-type: none"> ◆ librfccm.o ◆ libsapjcorfc.so ◆ sapjco.jar

Platform	SAP JAVA Connector Files
HP-UX11	<ul style="list-style-type: none"> ◆ librfccm.sl ◆ libsapjcorfc.sl ◆ sapjco.jar <p>Important: For JCo version 2.1.6 on HP-UX11: Any messages to stderr in the following format can be ignored: /usr/lib/dld.sl: Can't find path for shared library: dsrlib.sl /usr/lib/dld.sl: No such file or directory</p> <p>These messages have no impact on JCo functionality. This library is used for debugging the sapjco libs (SAPCCMSR). However, if you want to use this library, you can download it from SAP.</p>
Linux	<ul style="list-style-type: none"> ◆ librfccm.so ◆ libsapjcorfc.so ◆ sapjco.jar
Sun Solaris	<ul style="list-style-type: none"> ◆ librfccm.so ◆ libsapjcorfc.so ◆ sapjco.jar
Microsoft Windows	<ul style="list-style-type: none"> ◆ librfc32.dll ◆ sapjco.jar ◆ sapjcorfc.dll <p>Important: For JCo version 2.1.6 or later on Windows: If you are using the SAP JCo version 2.1.6 or later, you must follow the instructions in SAP Note 684106, which explains that the following Microsoft runtime DLLs on your Windows system need to be updated:</p> <ul style="list-style-type: none"> ◆ msucr71.dll ◆ msvcp71.dll ◆ mfc71.dll ◆ mfc71u.dll <p>The files are attached to the SAP note in an installable archive. Unpack the archive, and then install using R3DLLINS.exe.</p>
IBM iSeries	<ul style="list-style-type: none"> ◆ sapjco.jar ◆ Librfc.savf ◆ Sapjcorfc.savf

Installing on UNIX

To install the SAP Java Connector on UNIX:

1. Complete the procedure *Downloading the SAP Java Connector* on page 1274.
2. Shut down Application if it is running.

3. To set the CLASSPATH and LIBPATH/LD_LIBRARY_PATH, complete the following steps:

a. For the operating system-specific libraries, type:

```
install3rdParty.sh sapjco 2_1_6 (or the most recent version within 2.1.x) -j  
directory/*.jar
```

b. For the native libraries, type:

```
install3rdParty.sh sapjco 2_1_6 (or the most recent version within 2.1.x) -l  
directory/*.so
```

Notes:

- ◆ Two different switches are used in the previous commands (-j and -l). Make sure you use the correct switch, as shown in the command.
- ◆ lib-extension depends on the operating system you are using. For Sun Solaris, it would be .so, for example. See the table that lists *SAP JAVA Connector Files* on page 1274.
- ◆ If you are using the MySQL database, ensure that the database is stopped after running `install3rdParty.sh`.

4. Restart Application.

Installing on Windows

To install the SAP Java Connector on Windows:

1. Complete the procedure described in *Downloading the SAP Java Connector* on page 1274.
2. Shut down Application if it is running.
3. To add the sapjco.jar and the native libraries to the Windows registry, complete the following steps:

Notes:

- ◆ Two different switches are used in these commands (-j and -l). Make sure that you use the correct switch.
- ◆ If you are using the MySQL database, ensure that the database is stopped after running `install3rdParty.cmd`.

a. For the operating system-specific libraries, type:

```
install3rdParty.cmd sapjco 2_1_6 (or the most recent version within 2.1.x) -j  
directory\*.jar
```

b. For the native libraries, type:

```
install3rdParty.cmd sapjco 2_1_6 (or the most recent version within 2.1.x) -l  
directory\*.dll
```

After typing this command, complete the following steps:

- 1) Run `stopWindowsService.cmd`.
- 2) Run `uninstallWindowsService.cmd`.
- 3) Close the Command window and re-open it.
- 4) Run `InstallWindowsService.cmd`.

- c. If you receive a database connection test failure error, in the \bin directory, start MySQL™ using the following command:

```
control_mysql.cmd start
```

4. Restart Application.

Installing on iSeries

To install the SAP Java Connector on iSeries, complete the following steps:

1. Complete the procedure described in *Downloading the SAP Java Connector* on page 1274.
2. From the jar directory for your Application installation, copy **sapjco.jar** into a working directory on your iSeries.

Note: If you use FTP to copy the files, make sure that you transfer the files in binary mode.

3. From the lib directory for your Application installation, copy **LIBRFC.SAVF** and **SAPJCORFC.SAVF** into a working directory on your iSeries.
4. Create a library to store the SAP service programs. From an iSeries command line, type **CRTLIB SAPJCOLIB**. If this library already exists, you can choose another name.
5. Prepare the service programs for the restore by copying them from the IFS to the QSYS side of your iSeries:

```
CPYFRMSTMF FROMSTMF('/copytodirectory/SAPJCORFC.SAVF')
TOMBR ('/QSYS.LIB/SAPJCOLIB.LIB/SAPJCORFC.FILE')
MBROPT (*REPLACE)
```

and

```
CPYFRMSTMF FROMSTMF('/copytodirectory/LIBRFC.SAVF')
TOMBR ('/QSYS.LIB/SAPJCOLIB.LIB/LIBRFC.FILE')
MBROPT (*REPLACE)
```

6. Restore the service programs to library **SAPJCOLIB** by typing:

```
RSTOBJ OBJ(*ALL) SAVLIB(M630SPREL) DEV(*SAVF)
SAVF(SAPJCOLIB/SAPJCORFC) RSTLIB(SAPJCOLIB)
```

and:

```
RSTOBJ OBJ(*ALL) SAVLIB(M630SPREL) DEV(*SAVF)
SAVF(SAPJCOLIB/SAPJCORFC) RSTLIB(SAPJCOLIB)
```

These two commands place the service programs **LIBRFC** and **SAPJCORFC** in the library **SAPJCOLIB**.

7. Edit the job description that you use to start your Application instance. Add the library **SAPJCOLIB** to the initial library list of that job description.
8. Add **SAPJCOLIB** (or a new name, as described in step 4) to the **LIBPATH** environment variable. Type the following command:

```
ADDENVVAR ENVVAR(LIBPATH) VALUE('/QSYS.LIB/SAPJCOLIB.LIB') LEVEL(*SYS)
```

9. Ensure that your Application instance is not running.
10. From an iSeries command line, type **QSH** and press **Return** to get into Qshell.

11. Change to the bin directory of your Application instance.
12. Type the following command:

```
install3rdParty.sh sapjco lz -j /locationofthesapjco.jarfile/sapjco.jar  
-nodeploy > i3p.log 2> i3p.log
```
13. Upon completion, type:

```
deployer.sh > deployer.log 2> deployer.log
```
14. Once that has completed, verify that the passphrase=password has not been removed from the Application *install_dir*/properties/security.properties file. If it has, add it back.
15. Restart Application.

Creating an SAP Suite Adapter Configuration

Before You Begin

Before you begin to configure the SAP Suite adapter, collect the following information:

- SAP Application Server IP address (non-load balanced SAP R/3 system)
- SAP Message Server (load balanced SAP R/3 system only)
- R/3 Name (load balanced SAP R/3 system only)
- SAP Group (load balanced SAP R/3 system only)
- SAP Gateway Host IP address (if applicable, required for SAP outbound processing)
- SAP Gateway Service (if applicable, required for SAP outbound processing)
- SAP System Number
- SAP Port (Mode File based IDoc RFC only)
- SAP Client Number
- SAP Program ID (SAP outbound processing only)
- Username for logging into the SAP Application Server
- Password for logging into the SAP Application Server
- Name of the business process to start (SAP outbound processing only)
- Name of the Remote Function Call(s) to register (SAP RFC synchronous requests and BAPI only. Required for SAP outbound processing only.)

See *Configuring the SAP Suite Adapter* on page 1279 for a description of the parameters used to define this information.

In addition, for file-based IDoc processing, you need to create an FTP login authentication file (required by the FTP server) and place the file in the directory where Application is installed. The file must contain the following values:

```
username=sapusername  
password=sappassword
```

Configuring the SAP Suite Adapter

Whether you plan to create a business process that includes the SAP Suite adapter or use the predefined business processes, you must create a service configuration of the SAP Suite adapter. For more information, see *Managing Services and Adapters*.

To create and enable a configuration of the SAP Suite adapter, use the following table to configure the parameters:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select a Service Group to associate with this adapter. Valid values: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. For more information on Service Groups see chapter 1, <i>Managing Services and Adapters</i> .
SAP Integration Mode (Mode)	Integration mode that enables Application to communicate with an SAP R/3 system and with transfer documents. Required. Valid values: <ul style="list-style-type: none">◆ Synchronous BAPI – Enable integrated activities using SAP R/3 business objects and their BAPIs, which are processed in synchronous mode.◆ Synchronous RFC – Trigger direct and synchronous RFCs that start SAP R/3 function modules or receive synchronous RFC calls from SAP R/3.◆ File-based IDoc RFC – Transfer documents using SAP R/3 IDoc technology when communicating with SAP R/3 using RFC.◆ ALE-based IDoc – Transfer documents using SAP R/3 IDoc technology when communicating with SAP R/3 using ALE.
SAP system is load balanced (LoadBalanced)	Whether Application is connecting to a single SAP R/3 system or cluster of SAP R/3 systems. Required. Valid values: <ul style="list-style-type: none">◆ Yes – Connect to a cluster of SAP R/3 systems.◆ No – Connect to a single SAP R/3 system.

The integration mode and load-balancing determine which of the following parameters display for configuration. Refer to the following sections to determine the information that you must provide to continue creating the SAP Suite adapter service configuration:

Synchronous BAPI on page 1280

Synchronous RFC on page 1281

File-based IDoc RFC on page 1281

ALE-based IDoc RFC on page 1282

User Properties on page 1283

RFC Server Configuration (Outbound) on page 1284

Connection Pool Settings on page 1285

Application User Settings on page 1286

Synchronous BAPI

The following table describes the fields to configure for synchronous BAPI:

Field	Description
Load Balanced	
SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the SAP R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Not Load Balanced	
SAP Application Server	Name of the server for the SAP R/3 application. Required.
Gateway Host (GWhost)	Host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (GWserv)	Gateway service of the SAP R/3 system. Optional.
SAP System Number	Number of the SAP R/3 system. Required.

Synchronous RFC

The following table describes the fields to configure for synchronous RFC:

Field	Description
Load Balanced	
SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the SAP R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Gateway Host (GWhost)	Host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (GWserv)	Gateway service of the SAP R/3 system. Optional.
Not Load Balanced	
SAP Application Server	Name of the server for the SAP R/3 application. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use your IP address. Optional.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.
SAP System Number	Number of the SAP R/3 system. Required.

File-based IDoc RFC

The following table describes the fields to configure for file-based IDoc RFC:

Field	Description
Not Load Balanced	
SAP Application Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.
SAP System Number	Number of the SAP R/3 system. Required.

Field	Description
SAP Port (Port)	Port or communication type that enables the SAP R/3 system to communicate with external systems during electronic data interchange. For example, for EDI subsystems that read IDocs in the form of sequential files, the SAP R/3 port is File. Required.
Map SAP IDoc path to local path	Enable Application to map the local directory path to the IDoc on the SAP R/3 system.
Local Path (MapPath)	Local directory path to the IDoc on the SAP R/3 system. Required if you selected the Map SAP IDoc path to local path check box.
Load Balanced	
SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Optional.
SAP Port (Port)	Port or communication type that enables the SAP R/3 system to communicate with external systems during electronic data interchange. For example, for EDI subsystems that read IDocs in the form of sequential files, the SAP R/3 port is File. Required.
Map SAP IDoc path to local path	Enable Application to map the local directory path to the IDoc on the SAP R/3 system.
Local Path (MapPath)	Local directory path to the IDoc on the SAP R/3 system. Required if you selected the Map SAP IDoc path to local path check box.

ALE-based IDoc RFC

The following table describes the fields to configure for ALE-based IDoc RFC:

Field	Description
Not Load Balanced	
SAP Application Server	Name that identifies the application server for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Required.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.

Field	Description
SAP Version (SAPVersion)	Version of the SAP R/3 system to control IDoc version. Required. Valid values are: <ul style="list-style-type: none"> ◆ 3 – IDoc version used in SAP releases 3.x (EDI_DC) ◆ 4 – IDoc version used in SAP releases 4.x (EDI_DC40)
Check box Retry Sending IDocs	Check if SAP Suite adapter should automatically retry to deliver the IDoc to the SAP system.
Max Retries	Sets how often the retry should take place. Setting the value to 0 causes the SAP Suite adapter to retry infinitely.
Retry Sending Interval	Delay in seconds between two retries.

Load Balanced

SAP Message Server	Name that identifies the message server for the cluster of SAP R/3 systems. Required.
R/3 Name (R3name)	Name of the SAP R/3 system. Required.
SAP Group (Group)	Group name that is configured for the cluster of SAP R/3 systems. Required.
Gateway Host (Gwhost)	Name of the host running the SAP R/3 gateway server. You can use an IP address. Optional.
Gateway Service (Gwserv)	Gateway service of the SAP R/3 system. Optional.
SAP Version (SAPVersion)	Version of the SAP R/3 system. Required.
Check box	Check if SAP Suite adapter should automatically retry to deliver the IDoc to the SAP R/3 system.
Max Retries	Sets how often the retry should take place. Setting the value to 0 causes the SAP Suite adapter to retry infinitely.
Retry Sending Interval	Delay in seconds between two retries.

User Properties

The following table describes the fields to configure user properties:

Field	Description
Client (Client)	Name of the client in the SAP R/3 system. Required.
User (User)	Name of the authorized user of the SAP R/3 account. Required.
Password (Passwd)	Password of the authorized user of the SAP R/3 account. Required.

Field	Description
Language (Lang)	Language used by the SAP R/3 account. Required. Valid value is EN (English).
Code Page (Codepage)	Code page of this SAP R/3 system—for example, 1100 for English and most European languages. Optional.

RFC Server Configuration (Outbound)

The following table describes the fields to configure RFC properties:

Field	Description
Start RFC Server automatically	Start the local RFC server automatically. The RFC server is required for SAP R/3 outbound.
RFC Server Instances	Number of RFC server instances listening for connections from the SAP R/3 system.
Program ID (Program ID)	The SAP system program ID used to tie the SAP Suite adapter instance to the correct SAP system for handling outbound requests.
Outbound Process to Start	Select the name of the business process that Application should run after receiving an IDoc from SAP R/3.
Outbound Encoding	Set the character encoding for the outbound document.
Wait for synchronous RFC Outbound response	Enables support for synchronous responses in case of outbound SAP RFC calls.
Response Timeout	After this timeout (in seconds), the RFC server closes the connection to the SAP system. After this timeout, if Application wants to send a response back to SAP, an error is returned.
Delete TIDs Automatically	Indicates whether transaction IDs (TIDs) that have been already used should be deleted after the time configured in the Delete TIDs after (days) field. Note: TIDs are used for ALE processing only.
Delete TIDs after (days)	TIDs older than <i>n</i> days are deleted.

Field	Description
Register Remote Function	<p>This applies to BAPI or RFC Outbound mode only. To be able to handle outbound BAPI or RFC calls, the call structures have to be registered in Application.</p> <p>Options are:</p> <ul style="list-style-type: none"> ◆ Select RFCs online from RFC list Application establishes a connection to SAP and retrieves the list of available RFC or BAPI functions. ◆ Enter RFCs offline A comma-separated list of RFCs can be entered offline; no connection to the SAP system is established. ◆ None (for IDoc modes or BAPI/RFC Inbound): No functions need to be registered.
Checkbox SAP R/3 is a UNICODE system	<p>To be able to interface with Unicode SAP systems, you must enable the check box.</p> <p>Note: This option makes it unnecessary to add the line "sap.(instancename).Unicode = 1" to the sap.properties file, which was necessary in earlier versions of the SAP Suite adapter:</p>

Note: For Unicode SAP systems:

To be able to interface with Unicode SAP systems, you must set the Unicode option for your RFC destination in transaction SM59 on the SAP system (please contact the SAP system administrator for more information). You can find this option on the **Special Options** tab, in the **Character Width in Target System** section.

Connection Pool Settings

There are two types of connections to an SAP system: connections (physical) and sessions (logical).

Connections: Physical connections to an SAP system. Connections can be in either closed or open status.

Sessions: Logical connections to an SAP system. Each session object contains a connection object. A session can be in either used, unused, or called status.

The following table contains the settings used to configure a connection pool:

Field	Description
Use hard max connections limit	Check this box if the number of open parallel connections should never exceed the number of configured maximum connections as configured in Max Connections (Off = soft limit).
Max Connections	Maximum number of open parallel sessions.
Soft Limit Delay Time (seconds)	After this time, if all max sessions are still used and a call is waiting to be executed, an additional session is created.
Connection Check Interval (seconds)	Interval in which the session and connection state is checked by the adapter.

Field	Description
Connection Idle Time (seconds)	If a session is unused for this period of time, the connection is closed.
Close session after maximum session time	Whether the adapter should close the connections after the Max Session Time (see next field). Select this option to prevent connections from staying opened indefinitely.
Max Session Time (minutes)	Maximum time after a session is not processed (if enabled).

Note: If invalid data (like ABC or 13 . 45) is entered in a pool setting, the setting uses its default value.

Application User Settings

The following table contains the user setting for the service configuration:

Field	Description
User ID	Name of the user to associate with the business process that includes this service configuration.

You are now ready to complete either of the following tasks:

- Create a business process that includes the SAP Suite adapter.
- Configure SAP R/3 business processes for implementation (see the next section).

Editing the SAP Suite Adapter Configuration Parameters

Upon installation, the connection pool, RFC server, and trace parameters that enable connection to an SAP system are preconfigured.

To edit these parameters:

1. From the **Administration** menu, select **Deployment > Services > Installation/Setup**.
2. Under List, select **S** next to Alphabetically and click **Go!**

Note: Under List for Service Status, Installed (default value) should display.

3. From the list of services, select **edit** next to SAP Suite adapter.
4. Using the following table, edit the fields as appropriate:

Field	Description
Max. Startup Delay (MaxStartupDelay)	Maximum number of startup connections to the SAP R/3 system. Required. The default value is 60 seconds.

Field	Description
RFC Trace (RFCTrace)	RFC trace processing information about connections in the pool. Valid values: <ul style="list-style-type: none"> ◆ On – Trace the connection. ◆ Off – Do not trace the connection.
JCO Trace Level (JCOTraceLevel)	JCo trace processing information level. Valid values are 0 to 6 (6 is the maximum level).
JCO Trace Path (JCOTracePath)	Directory for the JCo trace output files. If “.” is used, the trace files will be located in the application server installation directory.

5. Click **Next** and review the parameters that you edited.
6. Click **Finish** to update Application.

Implementing the SAP R/3 Business Processes

To implement the business processes for SAP R/3, you need to perform the minimal configurations for the SAP Outbound IDoc, SAP Inbound Delivery, and SAP Delivery ALE business processes. The SAP Inbound IDoc and SAP Outbound ALE business processes require no additional setup.

To implement the business processes:

1. Check out the following predefined business processes (.bp files) from Application:
 - ◆ SAPInbDelivery.bp (for file-based IDoc)
 - ◆ SAPOutboundIDoc.bp (for file-based IDoc)
 - ◆ SAP Delivery ALE (for ALE IDoc)
2. After checking out the business processes, use the element editors in the GPM to display the configuration parameters within each business process.

Refer to the following sections to determine the information that you must provide to implement the business processes.

SAP Outbound IDoc (SAPOutboundIDoc.bp)

The following table provides the parameters to define for the preconfigured services, as appropriate:

Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the FTP Send adapter has entered the retirement process in Application and will be

replaced with the FTP Client Adapter with related services. For more information about the retirement process, see *Retiring and Removed Services and Adapters*.

Service/Adapter Configuration Instance Name	Parameter	Description
FTPSend (FTP Send Adapter)	xport-ftp-document	Name of IDoc that Application retrieves from SAP R/3. Required.
	xport-ftp-host	IP address or host name of the external trading partner host system. Valid values are valid IP addresses and host names. Required.
	xport-tp-authfile	Authentication file containing the user name, password, and passphrase. Valid value is the file name. Required if passphrase is used.
DocumentExtractionService (Document Extraction Service)	BatchLikeDocuments	Whether to split the IDocs that are extracted into batches. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes – Group IDocs into files based on Sender ID, Receiver ID, and Acceptor Lookup Alias. ◆ No – Split out each IDoc individually, regardless of Sender ID, Receiver ID, and Acceptor Lookup Alias values.
	DocExtractMapList	Name of the map to extract documents from a single batch file. Required if splitting IDocs into batches.
FS_WriteEDI (File System Adapter)	assignedFilename	Unique file name used to overwrite the file name created by the business process. If the file name is not unique, the previous file with the same name is overwritten the next time it runs. Valid value is any valid file name. Required.
	extractionFolder	Any folder or subfolder on the same computer where Application is installed and where it extracts (writes) data as part of a business process. Required.

SAP Inbound Delivery (SAPInbDelivery.bp)

The following table provides the parameters to define for the preconfigured services, as appropriate:

Note: Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the FTP Send adapter has entered the retirement process in Application and will be

replaced with the FTP Client Adapter with related services. For more information about the retirement process, see *Retiring and Removed Services and Adapters*.

Adapter Configuration Instance Name	Parameter	Description
FTPSend (FTP Send Adapter)	xport-ftp-dir	Folder name or mailbox ID of an external trading partner. Required.
	xport-ftp-document	Document that the trading partner within Application retrieves from the remote trading partner. Optional.
	xport-ftp-host	IP address or host name of the external trading partner host system. Valid values are valid IP addresses and host names. Required.
	xport-tp-authfile	Authentication file containing the user name, password, and passphrase. Valid value is the file name. Required if passphrase is used.
SapSuite (SAP Suite Adapter)		Configure the parameters for this preconfigured instance of the SAP Suite adapter or specify your custom SAP Suite adapter configuration instance. See <i>Creating an SAP Suite Adapter Configuration</i> on page 1278.

SAP Delivery ALE (SAPALEDelivery.bp)

The following table provides the parameters to define for the preconfigured services, as appropriate:

Adapter Configuration Instance Name	Parameter	Description
SapSuite (SAP Suite Adapter)		Configure the parameters for this preconfigured instance of the SAP Suite adapter or specify your custom SAP Suite adapter configuration instance. See <i>Creating an SAP Suite Adapter Configuration</i> on page 1278.

You are now ready to check in the business processes to Application.

Configuring an SAP R/3 Route

Application uses SAP routes to determine how to route IDocs to and from external trading partners. When creating the inbound routes, indicate which key fields in the IDoc EDI_DC control record are used to identify the IDoc.

Configuring an Inbound Route

To configure an inbound route:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Routes**.
2. Under Create, next to **New Inbound Route**, click **Go!**
3. Complete the fields in the following table, as appropriate:

Field	Description
Route Name	Unique, meaningful name for the inbound route. Required.
SAP Direction	Direction of this route. This value is read-only and is set to inbound. No configuration necessary.
SAP IDOC Version	Version of the IDoc that you are exchanging with SAP R/3. Required.
EDI Port Name	Name of the port that enables communication between Application and an SAP EDI subsystem. The EDI subsystem triggers the SAP Suite adapter when sending IDocs to Application. Application triggers the EDI subsystem when sending IDocs to SAP. Required. Note: Both the SAP port name and EDI port name are used to route the IDoc to the correct trading partner. These values, along with the SAP client number, are added to the IDoc control record during the final translation of the IDoc in the SAPInboundIDoc business process.
SAP Port Name	Name of the SAP internal port that enables communication between the EDI subsystem and the SAP system when sending and receiving IDocs. Required. Note: Both the SAP port name and EDI port name are used to route the IDoc to the correct trading partner. These values, along with the SAP client number, are added to the IDoc control record during the final translation of the IDoc in the SAPInboundIDoc business process.
SAP Client Number	SAP R/3 client number. Required.

4. Click **Next** and review your configuration settings.
5. Click **Finish** to add the inbound SAP R/3 route to Application.

Configuring an Outbound Route

To configure an outbound route:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Routes**.
2. Under Create, next to **New Outbound Route**, click **Go!**

3. Complete the fields in the following table, as appropriate:

Field	Description
Route Name	Unique, meaningful name for the outbound route. Required.
SAP Direction	Direction of this route. This value is read-only and is set to outbound. No configuration is necessary.
SAP IDOC Version	Version of the IDoc that you are exchanging with SAP R/3. Required.
EDI Port Name	<p>Name of the port that enables communication between Application and an SAP EDI subsystem. The EDI subsystem triggers the SAP Suite adapter when sending IDocs to Application. Application triggers the EDI subsystem when sending IDocs to SAP. Required.</p> <p>In addition, the EDI port name is included in the IDoc header and used along with the SAP port name and Acceptor Lookup Alias defined for the EDI envelope to identify the inner-most EDI envelope (which contains the translation map) to apply to the document. You must specify the EDI port name as the Receiver ID for the inner-most EDI envelope (for example, ST SE envelope).</p>
SAP Port Name	<p>Name of the SAP internal port that enables communication between the EDI subsystem and the SAP system when sending and receiving IDocs. Required.</p> <p>In addition, the SAP port name is included in the IDoc header and used along with the EDI port name and Acceptor Lookup Alias defined for the EDI envelope to identify the inner-most EDI envelope (which contains the translation map) to apply to the document. You must specify the SAP port name as the Sender ID for the inner-most EDI envelope (for example, ST SE envelope).</p>
SAP Partner Keys	<p>The key fields from the IDoc that Application uses to identify the IDoc. Required. Select all the key fields that apply.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ MESCOD – Logical message variant ◆ MESFCT – Logical message function ◆ MESTYP – Logical message type ◆ RCVPFC – Partner function of receiver ◆ RCVPRN – Partner number of receiver ◆ RCVPRT – Partner type of receiver ◆ SNDPRN – Partner number of the sender ◆ SNDPRT – Partner type of the sender ◆ STDMES – EDI message type ◆ TEST – Test option

4. Click **Next** and review your configuration settings.

5. Click **Finish** to add the outbound cross-reference to Application.

Configuring an SAP R/3 Cross-Reference

To enable Application to process inbound and outbound IDocs and translate them to and from EDI, you must specify SAP cross-references to look up the EDI envelope associated with the SAP route defined for the IDoc.

Note: The Application supports only one-to-one ratio of SAP cross-references to EDI envelopes.

Configuring an Inbound Cross-Reference

To configure an inbound cross-reference:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Route X-REF**.
2. Under Create, next to New Inbound X-REF, click **Go!**
3. Complete the fields in the following table, as appropriate:

Field	Description
Route Selection	Name of the inbound route to which you are creating this cross-reference. Required.
Envelope Selection	EDI envelope that you have previously created to translate the key field or fields that you specified when creating an inbound route. Required. Note: Specify the inner-most EDI envelope in this field.

4. Click **Next** and review your configuration settings.
5. Click **Finish** to add the inbound cross-reference to Application.

Configuring an Outbound Cross-reference

To configure an outbound cross-reference:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Routes > SAP Route X-REF**.
2. Under Create, next to New Outbound X-REF, click **Go!**
3. Complete the fields in the following table, as appropriate:

Field	Description
Selection Route	Name of the outbound route to which you are creating this cross-reference. Required.
Select Envelope	EDI envelope that you have previously created to translate the key field or fields that you specified when creating an outbound route. Required. Note: Specify the inner-most EDI envelope in this field.

4. Click **Next** and review your configuration settings, and then click **Finish** to add the outbound cross-reference to Application.

Configuring for Load Balancing

If the SAP system is load-balanced (“SAP system is loadbalanced” = Yes in the SAP Suite adapter configuration), you must configure the /etc/services file as shown in the following table. The /etc/services file is used to map port numbers to service names.

The location of this file depends on your host system (UNIX/Linux, Windows, or iSeries). In UNIX, the file is at /etc/services. In Windows, the file is at c:\WINDOWS\system32\drivers\etc\services. In iSeries, use the iSeries Transaction WRKSRVTBLE to edit this information.

Service Type	Port	Service Name
sapmsS01	3600/tcp	SAP System Message Port
sapdp00	3200/tcp	SAP System Dispatcher Port
sapdp00s	4700/tcp	SAP System Dispatcher Security Port
sapgw00	3300/tcp	SAP System Gateway Central Instance Port
sapgw00s	4800/tcp	SAP System Gateway Security Port

For the sapmsXXX service type, XXX is the SAP System ID. For example, if the SAP System ID is E01, the entry should read as follows:

```
sapmsE01 35YY/tcp #SAP System Message Port
```

YY is the SAP System Number.

Retrieving IDoc and Schema Format Descriptions

IDoc and schema format descriptions can be obtained either by using the new SapSuiteBuilder interface or with the command line-based tool, CmdlineSAPSuiteBuilder.

Retrieving Descriptions Using the SAP Suite Builder Interface

To retrieve IDoc and schema format descriptions using the SAP Suite Builder interface:

1. From the **Administration** menu, select **Deployment > Adapter Utilities > SAP Suite Builder**.
2. Enter the parameters required to access and log in to the SAP system.
Application connects to the SAP system and retrieves a list of available IDocs, BAPIs (objects and methods), or RFCs.
3. Select the required format descriptions.
4. Click **Finish** on the summary screen.
5. Download the formats as .ddf or .xsd files.
6. Import these format descriptions into the Application Map Editor.

Note: The **IDoc Meta Data Builder > Generation Parameters** page now includes the parameter **Generate IDoc version in records**. If enabled, the generated IDoc Record tag includes the segment definition name (*SEGMENTDEF, <segment_type><segment_version>*). If not enabled, the

generated IDoc Record tag includes the segment type name only (*SEGMENTTYP*, *<segment_type>*). This was the default in earlier versions of the SAP Suite Builder.

Retrieving Descriptions Using the SAP Command Line Interface

This section contains information about retrieving IDoc and schema format descriptions using the SAP Command Line interface. The `CmdlineSAPSuiteBuilder` is a command line-based tool that is installed when the SAP Suite adapter is installed. The `CmdlineSAPSuiteBuilder` enables a connection to an SAP R/3 system and delivers a particular IDoctype or schema (or its extension) in the form of a data definition format (.ddf) or schema (.xsd) file (see *DDF in create mode* on page 1296 or *XSD in create mode* on page 1297). In List mode, the `CMDLineSAPSuiteBuilder` delivers a list of either IDocs or schemas from the SAP R/3 system to which it is connected. You can then use the .ddf and the .xsd in the Application Map Editor to define your mapping requirements.

To use the utility, you should have a basic knowledge of:

- Types of IDocs (and their extensions) and RFCs (RFMs) in an SAP R/3 system
- Data formats (.ddf and .xsd)
- Data mapping concepts
- Application Map Editor

Requirements

Based on SAP R/3 requirements for retrieving IDocs, the `CMDlineSAPSuiteBuilder` utility needs particular RFC function calls, including `IDOCTYPE_READ_COMPLETE` and `IDOC_RECORD_READ`. Before running the utility, confirm that the necessary RFC function calls are available.

Note: The `IDOCTYPE_READ_COMPLETE` and `IDOC_RECORD_READ` RFC function calls are available in SAP release 4.5A (as standard) or 4.0B (with the additional support packages `SAPKH40B42` and `SAPKH40B48`).

Running the CmdlineSAPSuiteBuilder Utility

To run the `CmdlineSAPSuiteBuilder` utility, type one of the following commands:

UNIX:

```
bin/runSAPClass.sh schemaGenerator.CmdlineSAPSuiteBuilder <arguments>
```

Windows:

arguments

The `bin/runSAPClass.sh` or `bin\runSAPClass.cmd` script is a generic script that enables you to call any class (that has a main method) inside the SAP-related Application environment (that is, the path begins with `com.sterlingcommerce.Woodstock.services.sapsuite`). Therefore, the script expects, as a first parameter, the name of the Javaclass call (in this case, `schemaGenerator.CmdlineSAPSuiteBuilder`). The script may also call another external class file inside the `sapsuite` environment.

To see the syntax and usage information, type the command without *<arguments>*, with the following result:

Usage:

```
CmdlineSAPSuiteBuilder DDF <mode> <DDFBuilder-options>          or  
CmdlineSAPSuiteBuilder XSD <mode> <SchemaBuilder-options>
```

<mode> ::= -c create a DDF or XSD file (default)

|| -l list IDOCs or RFCs.

<DDFBuilder-options> - M)andatory or O)ptional - in create mode are:

```
-e <Extension>          O (CIMtype, e.g. ZORDERS01)  
-i <IDOCType>          M (Basicitype e.g. ORDERS01)  
-n                      O (no IDoc version in records will be generated)  
-o <outputfilename>    O (name of DDF-file to be generated)  
-p <propertyfilename>  M (file with SAP specific properties)  
-s <Segment_Release>  O (SAP Release (3 alphanum. chars, e.g. 40B) )  
-v <Record Type Vers> O (allowed values: 2 or 3 (default) )
```

<DDFBuilder-options> - M)andatory or O)ptional - in list mode are:

```
-f <Filterargument>    M (search for a list of IDCOS e.g. ORDERS*)  
-o <outputfilename>    O (name of listfile that keeps the found elements)  
-p <propertyfilename>  M (file with SAP specific properties)
```

-s <Segment_Release> O (SAP Release (3 alphanumeric chars, e.g. 40B))

<SchemaBuilder-options> - M)andatory or O)ptional - in create mode are:

-o <outputfilename> O (name of xsd-file to be generated

-p <propertyfilename> M (file with SAP specific properties)

-r <RFCname> M (RFCname whose Schema will be extracted)

<SchemaBuilder-options> - M)andatory or O)ptional - in list mode are:

-p <propertyfilename> M (file with SAP specific properties)

-f <Filterargument> M (search for a list of BO's, BAPI's or RFC's depending on searchtype)

-o <outputfilename> O (name of listfile that keeps the found elements

-t <searchtype> M (allowed values: BO or BAPI or RFC)

-a <additional arg> O (used in case of BAPI (M) or RFC (O))

There is always a create mode or a list mode for DDF and for XSD. The following examples illustrate how to use the different modes of the CmdlineSAPSuiteBuilder utility:

DDF in create mode

<arguments> =

```
DDF -c -i ORDERS01 -s 45A -v 3 -p SAPreadIdoc.properties -o orders01.ddf
```

Create the ddf file **orders01.ddf** that contains the IDoc structure of the IDOCTYPE ORDERS01 of segment release 45A and record type version 3. The corresponding SAP-related host and user information are read from the file SAPreadIdoc.properties.

The CmdlineSAPSuiteBuilder utility requests the password for the SAP username (field User) as configured in the property file (this is the case in every mode).

In addition, the CmdlineSAPSuiteBuilder program writes to a log file whose name is specified with the token saplogger.logfilename in the property file log.properties. Its amount of output ranges from only fatal messages to all messages (NONE, FATAL, ERROR, WARN, TIMING, INFO, DEBUG, ALL). Its value depends on the token saplogger.loglevel in the same property file log.properties that is common to the SAP Suite adapter package. This logging mechanism is in every mode.

DDF in list mode

<arguments> =

```
DDF -l -f ORDERS -s 45A -p SAPreadIdoc.properties
```

Deliver to stdout a list of all IDOCTYPES (and/or extensions) beginning with the string ORDERS.

XSD in create mode

<arguments> =

```
XSD -c -r BAPI_MATERIAL_AVAILABILITY -p SAPreadIdoc.properties -o  
BAPI_Material_Availibility.xsd
```

Create an XML schema from the RFC BAPI_MATERIAL_AVAILABILITY that, in this case, stands for a particular BAPI of the BusinessObject MATERIAL in the file BAPI_Material_Availibility.xsd. SAP-related properties are read from the file SAPreadIdoc.properties.

XSD in list mode

Using this, it is possible to filter for RFCs, BAPIs, or BOs, depending on the search type used:

Listing of RFCs

<arguments> =

```
XSD -l -t RFC -f BAPI_SALESORDER_CREATEFROMDAT -p SAPreadIdoc.properties
```

List all RFCs beginning with BAPI_SALESORDER_CREATEFROMDAT.

Listing of BOs

<arguments> =

```
XSD -l -t BO -f Mat -p SAPreadIdoc.properties
```

List all BusinessObjects beginning with the string “Mat.”

Listing of BAPIs

<arguments> =

```
XSD -l -t BAPI -a MATERIAL -f Get -p SAPreadIdoc.properties
```

List all BAPIs beginning with the string “Get” of the particular BusinessObject Material. Note that BAPIs can only be listed for one particular BusinessObject.

Property File Used by CmdlineSAPSuiteBuilder

The property file must be customized before using the utility. Each line in this file should have the following format:

<fieldname> = <value>

The following table describes the possible field names and their default values in the property file:

Note: All mandatory fields have no default value and therefore have to be set with the correct value. The range for the JCoTraceLevel is undocumented but the lowest level is 0 (no tracing).

Field Name	Default Value	Non LB, M or O/LB, M or O	Notes/Used Datatype
Gwhost		O	IP address or hostname
Gwserv		O	

Field Name	Default Value	Non LB, M or O/LB, M or O	Notes/Used Datatype
Client		M	Three-digit number
User		M	Maximum of 12 characters
Lang	EN	O	Exactly two characters
OutboundEncoding	UTF-8	O	Encoding used during generation of the DDF/schema definitions
Codepage	1100	O	Four-digit number
LoadBalanced	LB_OFF	O	If LB_OFF, the fields Ashost and Sysnr are mandatory. If LB_ON, the fields Mshost and R3name are mandatory.
Ashost		M / -	If not LoadBalanced: IP address or hostname
Sysnr		M / -	If not LoadBalanced: Number
Mshost		- / M	IP address of message server; if you used field name Ashost, this field is not used.
R3name		- / M	SAPSysID
RFCTrace	RFCTRACE_OFF	O	switch on: RFCTRACE_ON
JCoTraceLevel	0	O	Range: 0 - 9
JCoTracePath	.	O	

Error Messages

When the CmdlineSAPSuiteBuilder utility detects errors, the utility writes the errors to either stdout or to the logfile associated with saplogger.logfilename. Typically, errors are written to stdout only if during command-line parsing something was wrong. At this stage, the name of the logfile is still not known.

Business Process Definition Parameters –Transactions

The following table describes the business process parameters for transactions:

Parameter	Data Type/ HTML Type/ Validator/ Size/MaxSize	Value/ Default	Description
KeepSessionOpen	String/text/ Number	0,1	BAPI only: Causes the SAP Suite adapter to leave the connection open after executing the BAPI.
SessionID	String/text/	<sid>	BAPI only: Causes the SAP Suite adapter to connect to the SAP system using the session <sid> that was opened in a previous call. For information, see KeepSessionOpen (above).
BapiCommitWait	String/text	0, 1	BAPI only: BapiCommitWait is an optional parameter for a BapiCommit call. Causes the SAP Suite adapter to commit all pending calls and wait until the commit returns. (1=Wait, 0=Do not wait)
BapiCommit		0, 1	BAPI only: Causes the SAP Suite adapter to commit all pending calls without waiting. The result is returned in a primary document.
BapiRollback		0, 1	BAPI only: Causes the SAP Suite adapter to roll back all pending calls. The result is returned in a primary document.
RFCModuleName	String/text		RFC/BAPI only: The name of the remote function module or BAPI RFC to be run.
Encoding	String/text	UTF-8	Specifies the character set.
CreateTID	String/text	<tid>	RFC only: Open an SAP transaction and return a <tid> from SAP.
ConfirmTID	String/text	<tid>	RFC only: Confirms all RFC calls run in the current transaction <tid>.

Parameter	Data Type/ HTML Type/ Validator/ Size/MaxSize	Value/ Default	Description
IDocPathName	String/text		File-based IDoc RFC only: Specifies the name and path of the IDoc or status message from SAP system perspective.
IsStatusMessage	String/text/ Number	0,1	File-based IDoc RFC only: Indicates whether the transferred file is a status message. (0=IDoc, 1=status Message)
AutoCommit	String/text/ Number	0,1	BAPI only: Causes the SAP system to commit the BAPI call automatically after execution.
ServerResponse	String/text/ Number	0,1	The Server Response flag (ServerResponse = 1) indicates that a response document for a waiting RFC outbound should be returned to SAP. If the Server Response flag is set to 1, then the ServerSessionID and ServerSessionSequenceNumber must be passed to the SAP Suite instance to uniquely identify the waiting session with an RFC server instance.
ServerSessionID	String/text		The ServerSessionID uniquely identifies the destination RFC server instance of an SAP Suite instance for a server response. The ServerSessionID is generated by the RFC server and written into the process data of the started outbound business process.
ServerSessionSequence Number	String/text		The ServerSessionSequenceNumber uniquely identifies the session within the RFC server instance (identified by ServerSessionID) waiting for the response. The ServerSessionSequenceNumber is generated by the RFC server and written to the process data of the started outbound business process.

Business Process Definition Parameters

The following table describes the usage options of the business process parameters (IP = Instance Parameter, WP = Workflow Parameter):

Parameter	Type	Data Type/HTML Type/Validator/Size/MaxSize	Value/Default	sRFC	BAPI	RFC IDoc	ALE IDoc	Client or Server
CloseSession	WP	String/text/Number	0,1	O	O	-	-	C
KeepSessionOpen	WP	String/text/Number	0,1	O	O	-	-	C
SessionID	WP	String/text/	<sid>	O	O	-	-	C
BapiCommitWait	WP	String/text	0, 1	-	4.5: O 4.0: -	-	-	C
BapiCommit			0, 1		O			
BapiRollback			0, 1		O			
RFCModuleName	WP	String/text		M	M	-	-	C
Encoding	IP/WP	String/text	UTF-8	O	O	-	-	C
CreateTID	WP	String/text	<sid>	O	O	O	O	S
ConfirmTID	WP	String/text	<sid>	O	O	O	O	S
IDocPathName	WP/IP	String/text		-	-	M	-	S
IsStatusMessage	WP	String/text/Number	0,1	-	-	O	-	C
AutoCommit	WP	String/text/Number	0,1	-	O	?	O	C
Serverresponse	WP	String/text/Number	0,1	O	O			S
ServerSessionID	WP	String/text		O	O			S
ServerSessionSequenceNumber	WP	String/text		O	O			S

Export Parameters

The following table describes the business process parameters when exporting parameters:

Parameter	Data Type/HTML Type/Validator	sRFC	BAPI	RFC IDoc	ALE IDoc
SessionID	String	O	O	-	-

Parameter	Data Type/ HTML Type/Validator	sRFC	BAPI	RFC IDoc	ALE IDoc
TransactionID	String	O	O	-	-

TID Management for SAP Outbound

The SAP Suite adapter manages transactional integrity for SAP outbound to confirm that an IDoc packet has been successfully processed once. The SAP R/3 system assigns a transaction ID (TID) for every IDoc packet. The SAP Suite adapter stores the TID in the Application database in the SAP_TID table. Each row in the following table represents a separate IDoc packet and contains the following rows:

Row	Description
DATE_TIME	Date and time at which the TID table was updated.
TID (PK)	SAP transaction ID.
STATE	<ul style="list-style-type: none"> ◆ CREATED – Received the TID from SAP R/3 ◆ EXECUTED – Received the IDoc message with the TID and committed the transaction ◆ ROLLEDBACK – Rolled back the IDoc packet from further processing ◆ CONFIRMED – IDoc message has been queued for processing and the transaction has been committed
WFID	ID of the business process that has been started for processing the IDoc packet.
INSTANCE_NAME (PK)	Name of the service configuration of the SAP Suite adapter that received the IDoc packet.

It is important that entries in the TID table can only be modified by the service configuration of the SAP Suite adapter that created the entry. This includes the following guidelines:

Multiple RFC servers connected to the same Program ID can share the same TID management.

RFC servers from different configurations of the SAP Suite adapter (and different Program IDs) *cannot* share the same TID management.

The Outbound Flow Process

The outbound flow proceeds as follows:

1. SAP R/3 sends a TID to the service configuration of the SAP Suite adapter to which the RFC server is registered on the matching program ID.
2. The RFC server receives the TID and checks the TID table to determine whether it has previously received this TID from SAP R/3 or not. If the TID is not found in the TID table, then the SAP Suite adapter appends an entry to the TID file, specifying the date-time stamp, TID, and the state

(CREATED). The SAP Suite adapter returns a code to SAP R/3 indicating whether the TID was found, and the TID state determines whether SAP R/3 continues processing.

3. If SAP R/3 continues processing, the SAP Suite adapter starts a new transaction.
4. SAP R/3 sends the IDoc packet associated with the TID to the same service configuration of the SAP Suite adapter for this RFC server.
5. The SAP Suite adapter receives the IDoc packet and processes the data according to the way that is defined in the outbound business process, such as splitting IDoc packets or making routing decisions based on a map file.
6. After processing the data, the SAP Suite adapter returns success or an SAP exception (for example, if the target queue is full) to SAP R/3.
7. Based on the status returned from the SAP Suite adapter, SAP R/3 instructs the same service configuration to commit or roll back the transaction and performs one of the following actions:
 - ◆ The SAP Suite adapter commits the transaction and updates the date-time stamp and state (EXECUTED) in the TID file.
 - ◆ The SAP Suite adapter call rolls back the transaction and updates the date-time stamp and state (ROLLBACK) in the TID file.
 - ◆ If the transaction is successfully committed, the SAP Suite adapter updates the date-time stamp and state (CONFIRMED) in the TID file.

Advanced Status Returned by the SAP Suite Adapter

The following table includes the advanced status that may be returned by the SAP Suite adapter:

Status	Description
No Primary Document	No primary document was available when the SAP Suite adapter started running. The primary document is needed for BAPI, ALE, and RFC and contains the data to send to SAP R/3.
Cannot Commit and Rollback at the same time, please check your settings!	BAPI only. In the business process, both the BapiCommit and BapiRollback parameters are set to 1 at the same time, which is an invalid configuration.
Instance Parameter is missing	A service configuration of the SAP Suite adapter that is needed for this mode is missing.
Found no Session to close	BAPI only. While trying to actively close a session by using explicit business process parameters, the specified session in the Session ID parameter was not found. Check assignment of Session ID parameter.
No SessionID in Processdata found	The SAP Suite adapter started with a command that awaits the Session ID to be set in the business process.
Can not Create and Confirm a TID at the same time	RFC only. The SAP Suite adapter received instructions to create and confirm a TID (Transaction ID) at the same time. These instructions need to be separated into two different calls.

Status	Description
The TID value was empty when trying to confirm a TID!	FC only. The SAP Suite adapter received instructions to confirm a TID, but the Transaction ID Parameter was empty.
WrongSessionID	Connection session not found. Session ID may be expired.
OpenConnectionError	Error while establishing connection. The reason is explained in the report.
DisconnectError	Exception caught during disconnection.
NotConnectionPool	The RFC server is not initialized. The Connection pool will not be available.
OpenConnectionError	Error while establishing connection.
LoadNativeError	Cannot load native middleware library. Check library path and availability of native libraries.
JCOError	The library reported a fatal internal error. The native library may not be installed correctly.
NotMoreSession	Could not create another session (Session Limit reached).
ExitBecauseShutDown	No more sessions available; therefore, the SAP Suite adapter stops running.
RepositoryNotInited	The repository is not initialized.
GetTemplateError	Error when trying to retrieve a FunctionTemplate.
MetaDataNotAvailable	The metadata is not available in the repository.
CleanupError	Error during cleaning up JDBC connection.
RfcCallError	Error invoking the SAP RFC call.
ListenerError	Error in logging file listener.
SAPTableError	Error in analysis of SAP table.
RfcServerError	Error or Exception occurred in RFC server.
DisconnectError	Exception detected during disconnection.
PoolThreadError	Error in the connection pool test thread.
CreateTIDLogError	JDBC did not create any rows in Transaction Management table.
DeleteTIDLogError	JDBC did not delete a row in Transaction Management table.
LoadPropertyError	Error while loading properties from properties file.
ToDomConvertError	Error during the conversion to Document.
UpdateTIDLogError	JDBC did not update a row in Transaction Management table.
TidMaintenanceError	Error during maintenance of SAP_TID table.
OpenConnectionError	Error while establishing connection.

Status	Description
BapiDiscpatcherTransformer Exception	A problem occurred while mapping the Return Structure into process data.
BytesToDOMIOException	An IO Problem occurred while transferring Bytes into a DOM.
BytesToDOMParserException	The parser was not configured correctly while transferring bytes into a DOM.
BytesToDOMSaxException	A Sax problem occurred while transferring bytes into a DOM.
DomToBytesIOException	An IO problem occurred while transferring a DOM to bytes.
DomToBytesSaxException	A Sax problem occurred while transferring a DOM to bytes.
DomToStringIOException	An IO problem occurred while transferring a DOM to a string.
DomToStringSaxException	A Sax problem occurred while transferring a DOM to a string.
ParserConfigurationException	Parser configuration not property defined.
SENDALEAbapException	(SendALE) An Abap exception occurred.
SENDALEJcoException	(SendALE) A JCO exception occurred.

Running SAP Suite Adapter in an External JVM

The SAP Suite Adapter allows you to decide for each instance whether it should run within the Application JVM (standard behavior for all adapters) or in an external Java Virtual Machine (JVM) managed by an SAP Controller Server.

When you run SAP Suite Adapter instances external to Application, it provides stability to instance server and increases the critical IDoc size limit. For example, it does not affect the stability of Application, when large IDoc files are received. We recommend you to configure SAP Suite Adapter instances to run in an external JVM if you receive IDocs more than 2 MB (or 4000 IDoc segments) frequently in Application.

Configuring an SAP Suite Adapter in an External JVM

Adding SAP Controller Property

To setup SAP Suite Adapter instances to run in an external JVM, perform the following steps:

1. Add *SAPController* property to SAP Suite property file for each instance run externally. For example, `SAP.<sapsuite_instance_>.SAPController=External`

Note: Make sure that there are no spaces before and after “=” as shown in the example.

Property	Description
SAP.<sapsuite_instance_name>.SAPController	<p>(Optional) Indicates if the SAP Suite Adapter instance runs internally or externally.</p> <ul style="list-style-type: none"> ◆ Internal (default) – The SAP Suite Adapter instance runs in the Application JVM. ◆ External – The SAP Suite Adapter instance runs in an external JVM managed by a SAPController.

2. Install SAP Controller Windows Service for Windows.
3. Restart Application.

Installing SAP Controller Windows Service (Windows only)

You have to install SAP Controller once as a Windows service if you want to run SAP Suite Adapter instances in an external JVM. The SAP Controller is not installed automatically with *installWindowsService.cmd*, as it is an optional service.

Note: Install SAP JCo libraries before you install SAP Controller Service.

To install SAP Controller Windows service, perform the following steps:

1. Open command prompt.
2. Navigate to <siroot>\bin directory.
3. Type <siroot> and press Enter. The SAP Controller starts automatically after the installation is complete.

Starting the SAP Controller Server

The SAP Controller server should be started and registered to the Application JNDI context to start external SAP Suite instances in an external JVM.

Before starting the SAP Controller server, make sure that:

The SAP JCo libraries are installed on the SAP Controller server.

The SAP Controller server is started on the same machine where Application is installed.

If Application uses a 64-bit JVM, follow the steps described in *Modifying SAP Controller JVM Settings for 64-Bit Platforms* on page 1311 before starting SAP Controller (add -d64 JVM option).

Modify the security.properties file as described in *Modifying Security Properties* on page 1313 to make sure that Application passphrase is recognized when the SAP Controller starts.

If you have configured SAP Suite instances as *external* in sap.properties property file and required for Windows only, the SAP Controller Windows service is installed and then the server starts automatically by the standard Application scripts (UNIX) or (Windows). However, if you remove the external properties from sap.properties file and uninstall the SAP Controller Windows service on Windows, the SAP Controller will no longer start automatically with the Application scripts (UNIX) and (Windows).

When the SAP Controller is starting, it automatically starts all external SAP Suite instances configured as external with status as active. After the SAP Controller is started, the external SAP Suite instances are ready for inbound and outbound communication.

A file `sapcontroller.pid` is created in `<siroot>` directory indicating that SAP Controller is running. This file contains the process ID of the SAP Controller.

When the Application is started and SAP Suite instances are enabled, you can see the registered SAP Controller and one or more external SAP Suite instances with JNDI name suffix in the Application JNDI tree.

To view SAP Suite instances:

1. Login to Application.
2. Select **Operations > System > Troubleshooter**. The System Troubleshooting screen appears.
3. Select **JNDI Tree**. The SAP Suite instances appear along with other instances running in the system.

Name	ClassName
SAPController.node1	com.sterlingcommerce.woodstock.services.sapsuite.controller.SAPController
<instance>_SapSuite_node1.ext	com.sterlingcommerce.woodstock.services.sapsuite.SapSuiteServerImpl
<instance>_SapSuite_node1	com.sterlingcommerce.woodstock.services.sapsuite.SapSuiteServerImpl

Note: SAP Suite internal instances are also registered along with external instances.

Starting the SAP Controller Server Manually

To start the SAP Controller Server manually, perform the following steps:

1. Open command prompt.
2. Navigate to `siroot` directory.
3. Type the following command and press Enter.
 - ◆ For UNIX: `siroot`
 - ◆ For Windows: `siroot`

The SAP Controller Server starts and displays the following output:

```
SAP Controller started [PID=1958290], Log=<siroot>/logs/sapcontroller.log]
```

You can view the SAP Controller Service in the Windows Services dialog with the status *Started* as shown in the following figure. The port number may be different on your system.

Name	Description	Status	Startup Type	Log On As
Gentran Integration Suite WebDav at port 35300		Started	Manual	Local System
Gentran Integration Suite SAP Controller Server at port 35300		Started	Manual	Local System
Gentran Integration Suite Opsserver at port 35600			Manual	Local System
Gentran Integration Suite Opsserver at port 35300			Manual	Local System

Stopping the SAP Controller Server

The SAP Controller Server can be stopped by executing the following scripts:

UNIX: `< >/bin/stopSAPController.sh`

The file, `sapcontroller.pid` is deleted from the `<siroot>` directory indicating that SAP Controller Server is stopped.

Windows: `< >\bin\stopSAPControllerWindowsService.cmd`

You can also stop the SAP Controller service from the Windows Services dialog.

Enabling, Disabling, or Reconfiguring External SAP Suite Adapter Instances

The SAP Suite adapter instances running externally on an SAP Controller Server can be created, enabled, disabled, or reconfigured in the Application administration interface similar to the instances that run internally.

Enabling or disabling External SAP Suite Adapter Instances

The **System Troubleshooting > Adapters** screen displays both internal and external SAP Suite Adapter instances. If you disable an external SAP Suite adapter instance, the SAP Controller shuts down the instance and you will not be able to send or receive transmissions from the connected SAP system.

However, if you enable a previously disabled external SAP Suite adapter instance, the SAP Controller starts the instance again and reestablishes the connection to the SAP system.

Reconfiguring External SAP Suite Adapter Instances

If you change one or more global or instance properties of an external SAP Suite Adapter instance in the SAP Suite Adapter configuration menu and save the changes, the external adapter instance restarts with the new configuration properties automatically.

SAP Controller Logs

The SAP Controller creates a log file `sapcontroller.log` in the startup phase under `< >/logs` directory. After you register the SAP Controller in Application, it uses the standard SAP Suite log file `sap.log`. The SAP Controller log level is similar to the log level configured for SAP Suite Adapter.

Uninstalling the SAP Controller Windows Service (Windows only)

To uninstall SAP Controller Windows service, perform the following steps:

1. Open command prompt.
2. Navigate to `< >\bin` directory.
3. Type `uninstallSAPControllerWindowsService.cmd` and press **Enter**. The SAP Controller is uninstalled.

The SAP Controller is not uninstalled automatically with `uninstallWindowsService.cmd` as it an optional service.

Restarting SAP Controller Service

You can restart the SAP Controller Service by following the stop script steps as mentioned in Stopping the SAP Controller Server and start as mentioned in Starting the SAP Controller Server section.

When the SAP Controller is starting, it automatically starts all external SAP Suite instances configured as external with status as active. After the SAP Controller is started, the external SAP Suite instances are ready for inbound and outbound communication.

Checking SAP Controller Service Failover

The script `startSAPISAliveChecker [.cmd| .sh]` is a diagnosis tool to check for the SAP Controller Server state and the SAP Suite instances running in the controller.

You can call this script from the command line for the following purposes:

Check if the SAP Controller server has started successfully and registered in the Application JNDI context (default). The exit code of the script is 0 if the SAP Controller server is running and registered in the Application JNDI server or it is positive number if the SAP Controller has not started successfully.

List all SAP Suite Adapter instances started on SAP Controller Server (-l option).

Build a script to check the SAP Controller status and restart the SAP Controller automatically in situations when the SAP Controller JVM is terminated in an exception situation. For example, see the following script:

Unix:

```
<siroot>/bin/startSAPISAliveChecker.sh [-l [<count>]]  
                                     [-w <loop_wait_period>]  
  
startSAPISAliveChecker.sh -s
```

Windows:

```
<siroot>\bin\startSAPISAliveChecker.cmd [-l [<count>]]  
                                         [-w <loop_wait_period>]  
  
startSAPISAliveChecker.sh -s
```

Parameter	Description
-l <count>	(Optional) Executes <count> is-alive checks. <ul style="list-style-type: none">◆ If <count> is omitted, perform checks permanently.◆ If -l <count> is omitted then a single check is performed (same as -l 1) Default: 1 or permanently
-w <wait_period>	(Optional) Sets wait period between checks. Default: 10.000 ms
	(Optional) Prints list of SAP Suite instances started in SAP Controller Server

Restarting SAP Controller Server Example

The following scripts check the status of the SAP Controller and restart it if it is not active.

UNIX Example

The following script starts the SAP Controller server automatically if it is not active. It executes periodically by the standard UNIX scheduling mechanism crontab. The following crontab file checks the SAP Controller in an interval of 15 minutes each:

```
si_root
```

```
#!/bin/sh
# Restart SAP Controller Service if it is not active

cd <si_root>/install/bin
startSAPIsAliveChecker.sh
rc=$?
if [ $rc -ne 0 ]; then
    echo "The SAP Controller Server needs to be restarted [rc=$rc]"
    stopSAPController.sh
    startSAPController.sh
    exit 1
fi
```

Windows Example

The following script starts the SAP Controller server automatically if it is not active:

```
Windows, CheckSAPController.cmd
```

```
@echo off
setlocal
cd <si_root>\install\bin

call startSAPIsAliveChecker.cmd
set rc=%errorlevel%
if %rc% NEQ 0 (
    echo The SAP Controller Server needs to be restarted (rc=%rc%)"
    stopSAPControllerWindowsService.cmd
    startSAPControllerWindowsService.cmd
    exit /B %rc%
)
```

Modifying SAP Controller JVM Settings

You should never change the SAP Controller parameter default values unless you experience unusual situations like out-of-memory exceptions during operation. You can change values if you are asked to by Sterling Commerce support.

Modifying SAP Controller JVM Settings for 32-Bit Platforms

To change the JVM heap memory settings:

UNIX script - `startSAPController.sh.in`

1. Open `startSAPController.sh.in` in an Editor. For example, `vi`.
2. Search for the lines -

```
# Java heap memory settings
SAPCONTROLLER_JVMARGS="-Xms256m -Xmx1024m"
```
3. Change the heap size at startup (option `-Xms<min_size>`) or the maximum heap size (option `-Xmx<max_size>`) as instructed by Sterling Commerce Support.
4. Save the file and run `setupfiles.sh`.

Windows script -

1. Open command prompt.
2. Type `uninstallSAPControllerWindowsService.cmd` and press Enter.
3. Open `startSAPControllerWindowsService.cmd` in an Editor. For example, `vi`.
4. Search for the lines -

```
# Java heap memory settings
SAPCONTROLLER_JVMARGS="-Xms256m -Xmx1024m"
```
5. Change the heap size at startup (option `-Xms min_size`) or the maximum heap size (option `-Xmx max_size`) as instructed by Sterling Commerce Support.
6. Save the file and run `setupfiles.cmd`
7. Run `startSAPControllerWindowsService.cmd` to install the service with the modified JVM settings.

Modifying SAP Controller JVM Settings for 64-Bit Platforms

To change the JVM heap memory settings:

UNIX script -

1. Open `startSAPController.sh.in` in an Editor. For example, `vi`.
2. Search for the lines -

3. Add Java 64-Bit-Option “-d64” as follows -

4. Save the file and run setupfiles.sh.

Windows script -

1. Open command prompt.

2. Type `set JAVA_OPTS=-d64` and press **Enter**.

3. Open `setupfiles.sh` in an Editor. For example,

4. Search for the lines -

5. Add the following line before these lines -

6. Save the file and run setupfiles.cmd.

7. Run `setupfiles.cmd` to install the service with the modified JVM settings.

Heap Memory Consumption Example

The RFC call parameters are transferred over the network and collected in the RFC server component of the SAP Suite Adapter in heap memory. The RFC server component is part of the SAP JCo libraries, which consist of a Java layer and an underlying native code layer. The RFC Server allocates Heap Memory to store the actual RFC parameters transferred in the RFC call. This data size may be considerably higher than the sum of the used data in all RFC parameters (“raw” data size). The SAP Suite adapter always transfers the complete data structure with fixed maximum length records including the unused fields independent of whether the parameter is filled with data or not.

The following section analyses an example with an IDoc of 20 MB “raw” data size:

A MATMAS IDoc with 71300 lines (100 IDocs with 713 lines per Idoc) is transferred with the standard SAP outbound RFC call. Each line represents one IDOC_DATA_REC40 structure in SAP, which is 1063 bytes. You can compute the IDoc size in Java heap memory as follows:

In the RFC native layer, 75,791,900 bytes will be temporary allocated by “libRFC“ outside of the Java heap additionally. Therefore, a total of 227,375,700 bytes are allocated by the SAP JCo libraries. Additionally, the heap space allocated by the SAP Suite Adapter has to be added: The SAP JCo libraries pass the IDoc data to the SAP Suite Adapter in a data structure that does not contain unused data any more. While processing the IDoc data, the SAP Suite Adapter allocates about 3x raw data size heap memory, which is 3

x 20MB = 60MB. In the example a total of 227 MB + 60 MB = 287 MB is allocated on the heap for a “raw” 20 MB IDoc.

Layer	Memory Consumption of a 20MB Idoc [bytes]
Outside Java Heap - native layer	75,791,900
Java Heap - SAP JCo library	151,583,800 bytes
Java Heap - SAP Suite layer	60,000,000
Total	287,000,000

The following general formula computes the allocated heap space in bytes for an IDoc with “#idoc_lines” and a raw data size of “idoc_raw_size” bytes:

Layer	Computation Formula for Memory Consumption
Outside Java Heap - native layer	#idoc_lines * 1063 bytes
Java Heap - SAP JCo library	#idoc_lines * 1063 bytes * 2 bytes/Unicode-char
Java Heap - SAP Suite layer	3 * idoc_raw_size
Total	idoc_lines * 3189 bytes + 3 * idoc_raw_size

Modifying Security Properties

You can modify the security.properties file to make sure that Application pass phrase is recognized when the SAP Controller starts.

To modify the security.properties:

1. Open the properties/security.properties file.
2. At the end of the file, add the following line -

where

is the encrypted pass phrase of Application.

3. Save and close the file.

Note: Use the encrypt_string.sh tool to get the encrypted version of your Application pass phrase.

Usage Examples

The following sections contain examples using the SAP Suite adapter for inbound and outbound processing.

Inbound and Outbound IDoc Processing Preconditions

The inbound and outbound IDoc examples in the following sections focus only on the SAP Suite adapter configuration and the SAP inbound and outbound business processes for sending and receiving IDocs. To use the SAP inbound and outbound business processes to process EDI files, you need to perform the following additional steps:

1. Create maps that translate EDI to IDoc format (inbound processing) and IDoc format to EDI (outbound processing).
2. Create a business process for sending the translated IDoc to EDI file to the trading partner (outbound processing).
3. Define inbound and outbound EDI envelopes based on the EDI type (such as EDIFACT and X12) and:
 - ◆ Specify the translation maps for inbound and outbound processing.
 - ◆ Specify the SAP inbound business process for sending IDocs to SAP, and the business process for sending translated IDoc to EDI files to the trading partner.
4. Configure SAP inbound and outbound routes for the inbound and outbound IDocs. See *Configuring an Inbound Route* on page 1289 and *Configuring an Outbound Route* on page 1290.
5. Configure SAP inbound and outbound cross-references to tie the SAP inbound route and SAP outbound route to the appropriate trading partner EDI envelopes. See *Configuring an Inbound Cross-Reference* on page 1292 and *Configuring an Outbound Cross-reference* on page 1292.

Inbound Processing Examples

This section contains examples for the following inbound processing scenarios:

Sending an IDoc using Application Linking and Enabling (ALE) to an SAP system

Starting an SAP Business Application Programming Interface (BAPI) module to retrieve and return company information to Application

Sending an IDoc Using ALE

The following example illustrates an SAP Suite adapter configuration used for sending an IDoc document using ALE technology:

► **ExampleGPM SAP Suite ALE Inbound IDoc**

Service Settings

Service Type	SAP Suite Adapter
Description	Example for documentation
System Name	ExampleGPM SAP Suite ALE Inbound IDoc
Group	None
SAP Integration Mode	ALE based IDoc
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	None provided
Gateway Service	None provided
SAP System Number	99
SAP Version	4
Retry Sending IDocs	Yes
Max. Retries (0=unlimited)	5
Retry Sending Interval	10
Client	999
User	username
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	No
RFC Server Instances	1
Program ID	Ask_SAP_Administrator

Specifies the integration mode for interacting with the SAP system.

(Screen 1 of 2)

Outbound Process to start	None provided
Outbound Encoding	UTF8
Wait for synchronous RFC Outbound response	No
Response timeout (seconds, 0=unlimited)	0
Delete TIDs automatically	No
Delete TIDs after (days)	30
Register Remote Function Calls (BAPI or RFC Mode only)	None (for IDoc modes or BAPI/RFC Inbound)
Filter online RFC list by	None provided
Use hard max. connections limit (Off= soft limit)	No
Max. Connections	4
Soft Limit Delay Time (seconds)	120
Connection Check Interval (seconds)	30
Connection Idle Time (seconds)	240
Close session after maximum session time	Yes
Max. Session Time (minutes)	60
User	admin

(Screen 2 of 2)

The following example using the GPM illustrates a business process that uses the SAP Suite adapter to send an IDoc using ALE technology to an SAP system. The dimmed values were specified on the SAP Suite adapter configuration instance.

Service Editor - Send IDoc to SAP via ALE

Name: Send IDoc to SAP via ALE

Config: ExampleGPM SAP Suite ALE Inbound IDoc

Message To Service: Message From Service

Output Msg: Obtain Process Data first, then Messages

Message Name: SapSuiteInputMessage

Name	Value	Use XPAT
Ashost	127.0.0.1	<input type="checkbox"/>
AutoCommit		<input type="checkbox"/>
BapiCommit		<input type="checkbox"/>
BapiCommitWait		<input type="checkbox"/>
BapiRollback		<input type="checkbox"/>
BPName	[Not Applicable]	<input type="checkbox"/>
Client	999	<input type="checkbox"/>
CloseSession		<input type="checkbox"/>
Codepage	1100	<input type="checkbox"/>
ConfirmTID		<input type="checkbox"/>
ConnCheckInterval	30	<input type="checkbox"/>
ConnIdleTime	240	<input type="checkbox"/>
CreateTID		<input type="checkbox"/>

(Screen 1 of 4)

DeleteTIDAfterDays	30	<input type="checkbox"/>
Encoding		<input type="checkbox"/>
Group		<input type="checkbox"/>
Gwhost		<input type="checkbox"/>
Gwserv		<input type="checkbox"/>
HardLimit	No	<input type="checkbox"/>
IDocPathName		<input type="checkbox"/>
IsStatusMessage		<input type="checkbox"/>
JCOTraceLevel	0	<input type="checkbox"/>
JCOTracePath		<input type="checkbox"/>
KeepSessionOpen		<input type="checkbox"/>
Lang	EN	<input type="checkbox"/>
LoadBalanced	No	<input type="checkbox"/>
LocalDir		<input type="checkbox"/>
MapPath		<input type="checkbox"/>
MaxPoolSize	4	<input type="checkbox"/>
MaxRetry	5	<input type="checkbox"/>
MaxSessionTime	60	<input type="checkbox"/>

(Screen 2 of 4)

MaxStartupDelay	60	
Mode	ALE based IDoc	
Mshost		
OutboundEncoding	UTF8	
Passwd	*****	
Port		
ProgramID	Ask_SAP_Administrator	
R3name		
RetryInterval	10	
RFCFilter	RFC	
RFCList		
RFCModuleName		
RFCServerCount	1	
RFCTrace	Off	
RunServer	No	
SAPVersion	4	
SelectRFC	None (for IDoc modes or BAPIRFC Inbound)	
SendRetry	Yes	

Indicates that the IDoc is sent in ALE mode.

(Screen 3 of 4)

ServerResponse		
ServerSessionID		
ServerSessionSequenceNumber		
SessionID		
SessionMonitor	Yes	
SoftLimitDelayTime	120	
SyncWait	No	
SyncWaitTimeout	0	
Sysnr	99	
TIDDeletion	No	
TransactionID		
User	username	

(Screen 4 of 4)

The following example illustrates the same business process using BPML. The IDoc file orders.dat is the input passed to the business process and becomes the primary document. The primary document is input to the SAP Suite adapter.

```

<process name="ExampleSAPALEIDocInbound">
  <sequence name="Send IDoc">
    <operation name="Send IDoc to SAP via ALE">
      <participant name="ExampleGPM SAP Suite ALE Inbound IDoc"/>
      <output message="Xout">
        <assign to="." from="*" />
      </output>
      <input message="Xin">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>

```

SAP Suite adapter instance used to send the IDoc to SAP

The following example illustrates information returned from SAP to the business process, indicating that the SAP system transaction manager allowed Application to open a transaction:

```
<ProcessData>  
  <PrimaryDocument SCIOBJECTID="server11:754e616c:fb9c22589c:-70f" />  
  <TID>0A560DCB478E406B4D3502EF</TID>  
</ProcessData>
```

Transaction ID returned from the SAP system and placed in the process data of the initiating business process

Starting an SAP BAPI Module

The following example illustrates an SAP Suite adapter configuration used for starting a BAPI in an SAP system to retrieve company information.

► **ExampleGPM SAP Suite Inbound BAPI**

Service Settings

Service Type	SAP Suite Adapter
Description	Example for including in the documentation
System Name	ExampleGPM SAP Suite Inbound BAPI
Group	None
SAP Integration Mode	Synchronous BAPI
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	None provided
Gateway Service	None provided
SAP System Number	08
Client	999
User	uname
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	No
RFC Server Instances	1
Program ID	Ask_SAP_Administrator
Outbound Process to start	None provided
Outbound Encoding	UTF8

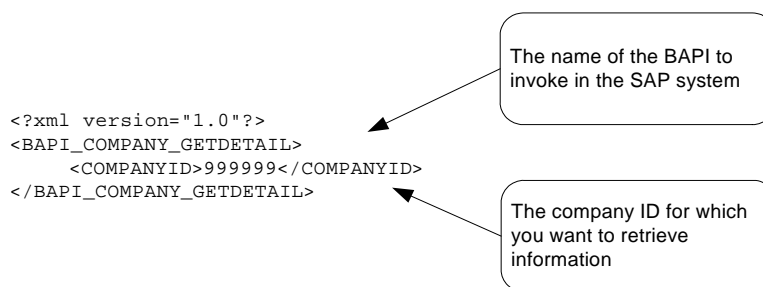
Specifies the integration mode for interacting with the SAP system.

(Screen 1 of 2)

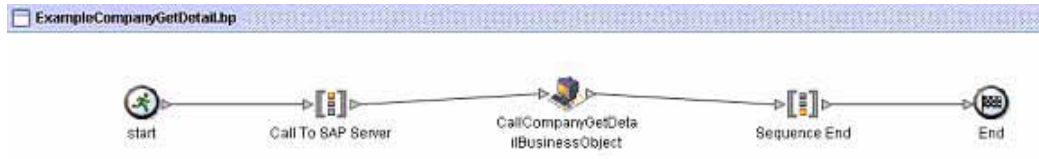
Wait for synchronous RFC Outbound response	No
Response timeout (seconds, 0=unlimited)	0
Delete TIDs automatically	No
Delete TIDs after (days)	30
Register Remote Function Calls (BAPI or RFC Mode only)	None (for IDoc modes or BAPI/RFC Inbound)
Filter online RFC list by	None provided
Use hard max. connections limit (Off = soft limit)	No
Max. Connections	4
Soft Limit Delay Time (seconds)	120
Connection Check Interval (seconds)	30
Connection Idle Time (seconds)	240
Close session after maximum session time	Yes
Max. Session Time (minutes)	60
User	admin

(Screen 2 of 2)

The following example illustrates the input passed to the business process.



The following example using the GPM illustrates a business process that uses the SAP Suite adapter to start the BAPI_COMPANY_GETDETAIL BAPI. The dimmed values were specified in the SAP Suite adapter configuration instance.



Service Editor - CallCompanyGetDetailBusinessObject

Name: CallCompanyGetDetailBusinessObject

Config: ExampleGPM SAP Suite inbound BAPI

Message To Service: Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: SapSuiteInputMessage

Name	Value	Use XPath?
Ashost	127.0.0.1	<input type="checkbox"/>
AutoCommit		<input type="checkbox"/>
BapiCommit		<input type="checkbox"/>
BapiCommitMail		<input type="checkbox"/>
BapiRollback		<input type="checkbox"/>
BPName	[Not Applicable]	<input type="checkbox"/>
Client	999	<input type="checkbox"/>
CloseSession		<input type="checkbox"/>
Codepage	1100	<input type="checkbox"/>

(Screen 1 of 5)

ConfirmTID		<input type="checkbox"/>
ConnCheckInterval	30	<input type="checkbox"/>
ConnIdleTime	240	<input type="checkbox"/>
CreateTID		<input type="checkbox"/>
DeleteTIDAfterDays	30	<input type="checkbox"/>
Encoding		<input type="checkbox"/>
Group		<input type="checkbox"/>
Gwhost		<input type="checkbox"/>
Gwserv		<input type="checkbox"/>
HardLimit	No	<input type="checkbox"/>
IDocPathName		<input type="checkbox"/>
IsStatusMessage		<input type="checkbox"/>
JCOTraceLevel	0	<input type="checkbox"/>
JCOTracePath	.	<input type="checkbox"/>
KeepSessionOpen		<input type="checkbox"/>
Lang	EN	<input type="checkbox"/>
LoadBalanced	No	<input type="checkbox"/>

(Screen 2 of 5)

LocalDir		<input type="checkbox"/>
MapPath		<input type="checkbox"/>
MaxPoolSize	4	<input type="checkbox"/>
MaxRetry	5	<input type="checkbox"/>
MaxSessionTime	60	<input type="checkbox"/>
MaxStartupDelay	60	<input type="checkbox"/>
Mode	Synchronous BAPI	<input type="checkbox"/>
Mshost		<input type="checkbox"/>
OutboundEncoding	UTF8	<input type="checkbox"/>
Passwd	*****	<input type="checkbox"/>
Port		<input type="checkbox"/>
ProgramID	Ask_SAP_Administrator	<input type="checkbox"/>
R3name		<input type="checkbox"/>
RetryInterval	10	<input type="checkbox"/>
RFCLiter	RFC	<input type="checkbox"/>
RFCList		<input type="checkbox"/>

Indicates synchronous BAPI mode.

(Screen 3 of 5)

RFCList			<input type="checkbox"/>
RFCModuleName	BAPI_COMPANY_GETDETAIL		<input checked="" type="checkbox"/>
RFCServerCount	1		<input type="checkbox"/>
RFCTrace	Off		<input type="checkbox"/>
RunServer	No		<input type="checkbox"/>
SAPVersion	4		<input type="checkbox"/>
SelectRFC	None (for IDoc modes or BAPIRFC Inbound)		<input type="checkbox"/>
SendRetry	Yes		<input type="checkbox"/>
ServerResponse			<input type="checkbox"/>
ServerSessionID			<input type="checkbox"/>
ServerSessionSequenceNumber			<input type="checkbox"/>
SessionID			<input type="checkbox"/>
SessionMonitor	Yes		<input type="checkbox"/>
SoftLimitDelayTime	120		<input type="checkbox"/>
SyncWait	No		<input type="checkbox"/>
SyncWaitTimeout	0		<input type="checkbox"/>

The name of the BAPI to invoke in the SAP system

(Screen 4 of 5)

Sysnr	08		<input type="checkbox"/>
TIDDeletion	No		<input type="checkbox"/>
TransactionID			<input type="checkbox"/>
User	uname		<input type="checkbox"/>

(Screen 5 of 5)

The following example illustrates the same business process using BPML.

```
<process name="ExampleCompanyGetDetail">
  <sequence name="Call To SAP Server">
    <operation name="CallCompanyGetDetailBusinessObject">
      <participant name="ExampleGPMSAPSuiteInboundBAPI"/>
      <output message="SAPOutput">
        <assign to="RFCModuleName" from="'BAPI_COMPANY_GETDETAIL'"/>
        <assign to="AutoCommit" from="0"/>
        <assign to="." from="*" />
      </output>
      <input message="SAPInput">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

The name of the BAPI to invoke in the SAP system.

The following example illustrates the company information returned from SAP to the business process as a primary document. The information is returned in XML format.

```
BAPI_COMPANY_GETDETAIL>
  <COMPANYID>999999</COMPANYID>
  <COMPANY_DETAIL>
    <COMPANY>999999</COMPANY>
    <NAME1>HANDSOME, INC</NAME1>
    <NAME2 />
    <COUNTRY>USA</COUNTRY>
    <LANGU>E</LANGU>
    <STREET>5555 EAST MARTIN AVE</STREET>
    <PO_BOX />
    <POSTL_COD1>80220</POSTL_COD1>
    <CITY>DENVER</CITY>
```

```

<CURRENCY>DOLLAR</CURRENCY>
<COUNTRY_ISO>USA</COUNTRY_ISO>
<CURRENCY_ISO>DOLLAR</CURRENCY_ISO>
<LANGU_ISO>USA</LANGU_ISO>
</COMPANY_DETAIL>
<RETURN>
<TYPE/>
<CODE/>
<MESSAGE/>
<LOG_NO/>
<LOG_MSG_NO>000000</LOG_MSG_NO>
<MESSAGE_V1/>
<MESSAGE_V2/>
<MESSAGE_V3/>
<MESSAGE_V4/>
</RETURN>
</BAPI_COMPANY_GETDETAIL>

```

In addition, the SAP Suite returns session information from the SAP system and puts it in the process data of the initiating business process. For example:

```

<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <PrimaryDocument SCIObjID="server1:252e596c:fb4e22589c:-76b0" />
  <BapiCallReturnStructure>
    <RETURN>
      <TYPE/>
      <CODE/>
      <MESSAGE/>
      <LOG_NO/>
      <LOG_MSG_NO>000000</LOG_MSG_NO>
      <MESSAGE_V1/>
      <MESSAGE_V2/>
      <MESSAGE_V3/>
      <MESSAGE_V4/>
    </RETURN>
  </BapiCallReturnStructure>
</ProcessData>

```

Outbound Processing Examples

This section contains examples for the following outbound processing scenarios:

- Receiving a file-based IDoc from an SAP system using RFC
- Receiving an ALE/IDoc from an SAP system using RFC
- Receiving a request from SAP and returning a synchronous response using RFC

Receiving a File-based IDoc from SAP Using RFC

This section includes an example SAP Suite adapter configuration and the predefined SAPOutboundIDoc business process that runs when a file-based IDoc is received from an SAP system.

The following example illustrates an SAP Suite adapter configuration used for receiving a file-based IDoc from an SAP system.

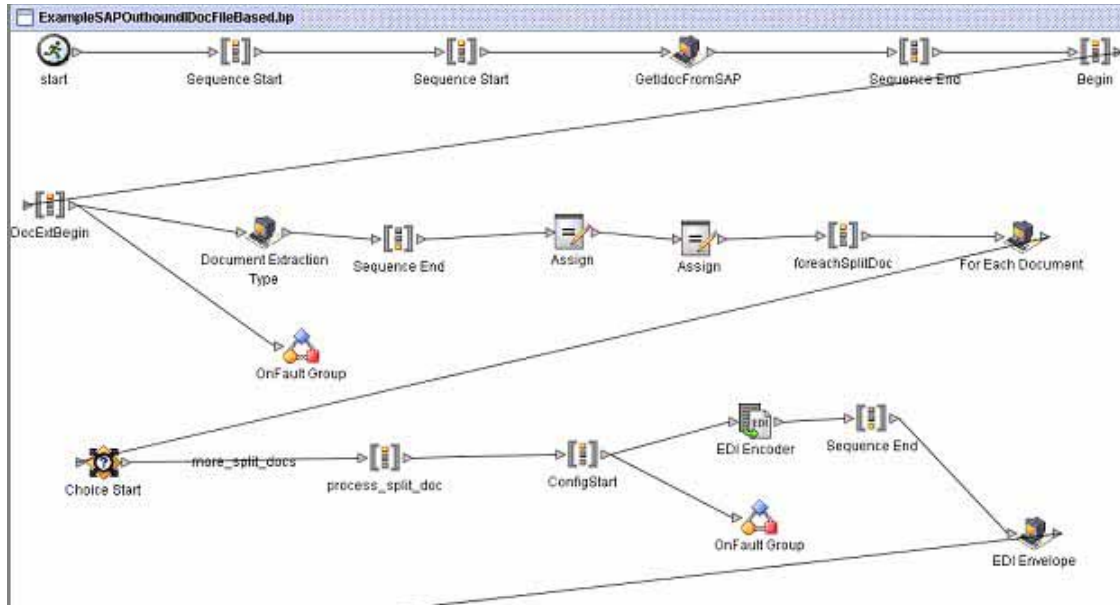
► **ExampleGPM SAP Outbound File Based IDoc**

Service Settings

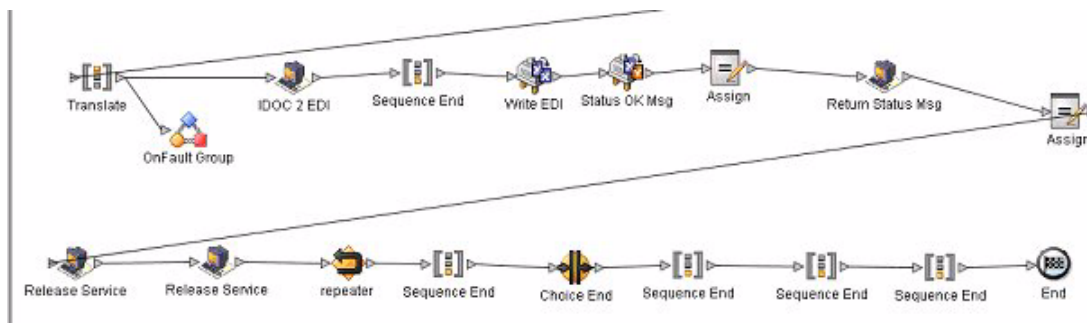
Service Type	SAP Suite Adapter	
Description	Configuration to include in documentation	
System Name	ExampleGPM SAP Outbound File Based IDoc	
Group	None	
SAP Integration Mode	File based IDoc RFC	Specifies the integration mode for interacting with the SAP system.
SAP system is loadbalanced	No	
SAP Application Server	127.0.0.1	
Gateway Host	127.0.0.1	
Gateway Service	sapgw01	
SAP System Number	01	
Client	999	
User	sapusername	
Password	*****	
Language	EN	
Code Page	1100	
Start RFC Server automatically	Yes	
RFC Server Instances	2	Business process invoked when a file-based IDoc is received from an SAP system
Program ID	SERVER_FRFC_CM	
Outbound Process to start	SAPOutboundIDoc	

The following example using the GPM illustrates the SAPOutboundIDoc business process that runs by the SAP Suite adapter for an outbound file-based IDoc. This business process retrieves the file-based IDoc from a directory on the SAP system and processes the file (translates the IDoc to EDI format and sends it to

a trading partner). In addition, the business process starts a subprocess that uses an instance of the SAP Suite adapter to send a status message back to the SAP system.



(Screen 1 of 2)



(Screen 2 of 2)

The following example illustrates the same business process using BPML.



```

<assign name="Assign" to="counter">1</assign>
<assign name="Assign" to="DOCUMENT_NAME_PREFIX">DOC-SPLIT-</assign>
<sequence name="foreachSplitDoc">
  <operation name="For Each Document">
    <participant name="ForEachDocument"/>
    <output message="ForEachDocumentTypeInputMessage">
      <assign to="DOCUMENT_NAME_PREFIX">DOC-SPLIT-</assign>
      <assign to="ITERATOR_NAME">DocExtract</assign>
      <assign to="." from="*" />
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
  <choice name="Choice Start">
    <select>
      <case ref="more_split_docs" activity="process_split_doc" />
    </select>
    <sequence name="process_split_doc">
      <sequence name="ConfigStart">
        <operation name="EDI Encoder">
          <participant name="EDI Encoder"/>
          <output message="EDIEncoderTypeInputMessage">
            <assign to="AcceptorLookupAlias" from="//*[name() =
concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/AcceptorLookupAlias/text()"/>
            <assign to="ReceiverID" from="//*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text())]/ReceiverID/text()"/>
            <assign to="SenderID" from="//*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text())]/SenderID/text()"/>
            <assign to="." from="*" />
          </output>
          <input message="inmsg">
            <assign to="." from="*" />
          </input>
        </operation>

```

Determines which envelope services need to run based on the AcceptorLookup Alias, SenderID, and ReceiverID.

Loops through each batch of IDocs and processes each document.


```

        <onFault>
            <sequence name="Sequence Start">
                <operation name="Status 04 Msg">
                    <participant name="Translation"/>
                    <output message="TranslationTypeInputMessage">
                        <assign to="." from="*"/>
                        <assign to="map_name">StatusMsg04</assign>
                        <assign to="PrimaryDocument" from="//
*[name() = concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/@SCIOBJECTID"/>
                    </output>
                    <input message="inmsg">
                        <assign to="." from="*"/>
                    </input>
                </operation>
                <assign name="Assign" to="IsStatusMessage">1</
assign>
                <operation name="Return Status Msg">
                    <participant name="InvokeSubProcessService"/>
                    <output
message="InvokeSubProcessServiceTypeInputMessage">
                        <assign to="INVOKE_MODE">ASYN</assign>
                        <assign to="NOTIFY_PARENT_ON_ERROR">ALL</
assign>
                    </output>
                    <assign to="WFD_NAME">SAPinbDelivery</
assign>
                    <assign to="." from="*"/>
                    </output>
                    <input message="inmsg">
                        <assign to="." from="*"/>
                    </input>
                </operation>
                <assign name="Assign" to="counter" from="counter +
1"/>
                <repeat name="repeater" ref="foreachSplitDoc"/>
            </sequence>
        </onFault>
    </sequence>
    <operation name="EDI Envelope">
        <participant name="EDIEnvelope"/>
        <output message="EDIEnvelopeTypeInputMessage">
            <assign to="MODE">IMMEDIATE</assign>
            <assign to="RECEIVER_ID" from="ReceiverID/text()"/>
            <assign to="SENDER_ID" from="SenderID/text()"/>
            <assign to="." from="*"/>
        </output>
        <input message="inmsg">
            <assign to="." from="*"/>
        </input>
    </operation>

```

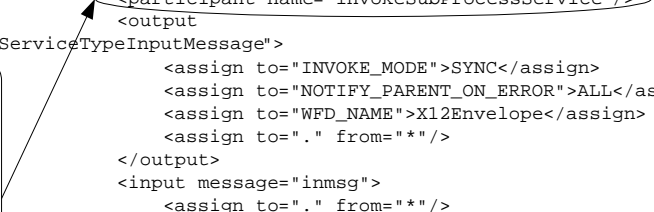
Determines which EDI enveloping business process needs to run to apply the EDI envelopes.

```

<sequence name="Translate">
  <operation name="IDOC 2 EDI">
    <participant name="InvokeSubProcessService"/>
    <output
message="InvokeSubProcessServiceTypeInputMessage">
      <assign to="INVOKE_MODE">SYNC</assign>
      <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
      <assign to="WFD_NAME">X12Envelope</assign>
      <assign to="." from="*" />
    </output>
    <input message="inmsg">
      <assign to="." from="*" />
    </input>
  </operation>
  <onFault>
    <sequence name="Sequence Start">
      <operation name="Status 05 Msg">
        <participant name="Translation"/>
        <output
message="TranslationTypeInputMessage">
          <assign to="." from="*" />
          <assign to="map_name">StatusMsg05</
assign>
          <assign to="PrimaryDocument"
from="//*[name() = concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/@SCIOBJECTID
"/>
          </output>
          <input message="inmsg">
            <assign to="." from="*" />
          </input>
        </operation>
        <assign name="Assign" to="IsStatusMessage">1</
assign>
        <operation name="Return Status Msg">
          <participant name="InvokeSubProcessService"/
>
          <output
message="InvokeSubProcessServiceTypeInputMessage">
            <assign to="." from="*" />
            <assign to="INVOKE_MODE">ASYN</assign>
            <assign
to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
            <assign to="WFD_NAME">SAPinbDelivery</
assign>
            </output>
            <input message="inmsg">
              <assign to="." from="*" />
            </input>
          </operation>
          <assign name="Assign" to="counter" from="counter
+ 1"/>
          <operation name="Release Service">
            <participant name="SAPRelease"/>
            <output
message="ReleaseServiceTypeInputMessage">
              <assign to="TARGET">/ProcessData/
*[local-name() = 'DOC']</assign>
              </output>
              <input message="inmsg">
                </input>
              </operation>
              <repeat name="repeater" ref="foreachSplitDoc"/>
            </sequence>
          </onFault>
        </sequence>
      </sequence>
    </onFault>
  </sequence>

```

Invokes the appropriate EDI enveloping business process to load the translation map and perform IDoc to EDI translation. After translation is complete, it invokes the business process specified in the envelope to complete processing and send the EDI file to the trading partner.



Writes the translated EDI document to the local file system.

```
<operation name="Write EDI">
  <participant name="FS_WriteEDI"/>
  <output message="FileSystemInputMessage">
    <assign to="assignedFilename" from="concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text())"/>
    <assign to="assignFilename">true</assign>
    <assign
to="extractionFolder">Enter_EDI_Extraction_Path</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
```

Constructs a status message to send back to the SAP system.

```
<operation name="Status OK Msg">
  <participant name="Translation"/>
  <output message="TranslationTypeInputMessage">
    <assign to="." from="*" />
    <assign to="map_name">StatusMsg</assign>
    <assign to="PrimaryDocument" from="/*[name() =
concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/@SCIOBJECTID
"/>
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
<assign name="Assign" to="IsStatusMessage">1</assign>
```

```

<operation name="Return Status Msg">
  <participant name="InvokeSubProcessService"/>
  <output
message="InvokeSubProcessServiceTypeInputMessage">
    <assign to="INVOKE_MODE">ASYNC</assign>
    <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
    <assign to="WFD_NAME">SAPInbDelivery</assign>
    <assign to="." from="*" />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
<assign name="Assign" to="counter" from="counter + 1"/>
<operation name="Release Service">
  <participant name="SAPRelease"/>
  <output message="ReleaseServiceTypeInputMessage">
    <assign to="TARGET">/ProcessData/*[local-name() =
'BATCH']</assign>
  </output>
  <input message="inmsg">
  </input>
</operation>
<operation name="Release Service">
  <participant name="SAPRelease"/>
  <output message="ReleaseServiceTypeInputMessage">
    <assign to="TARGET">/ProcessData/*[local-name() =
'GS']</assign>
  </output>
  <input message="inmsg">
  </input>
</operation>
<repeat name="repeater" ref="foreachSplitDoc"/>
  </sequence>
</choice>
</sequence>
</sequence>
</sequence>
</process>

```

Business process invoked to send a status message to the SAP system. The status message is in IDoc format.

The following example illustrates the status message sent to the SAP system upon successful completion of the outbound business process that ran.

```

EDI_D340 90000000000000692334      24      3      Control information of EDI subsystem OK
EDI_D340 90000000000000692334      05      3      Translation OK
EDI_D340 90000000000000692334      08      3      Syntax check OK
EDI_D340 90000000000000692334      10      3      Interchange handling OK

```

An SAP administrator can then view the status messages in the SAP system.

Receiving an ALE IDoc from SAP Using RFC

This section includes an example SAP Suite adapter configuration and the predefined SAPOutboundALE business process that runs when an IDoc is received from an SAP system using ALE technology.

The following example illustrates an SAP Suite adapter configuration used for receiving an IDoc from an SAP system using ALE Technology.

► ExampleGPMsAPOutboundALE

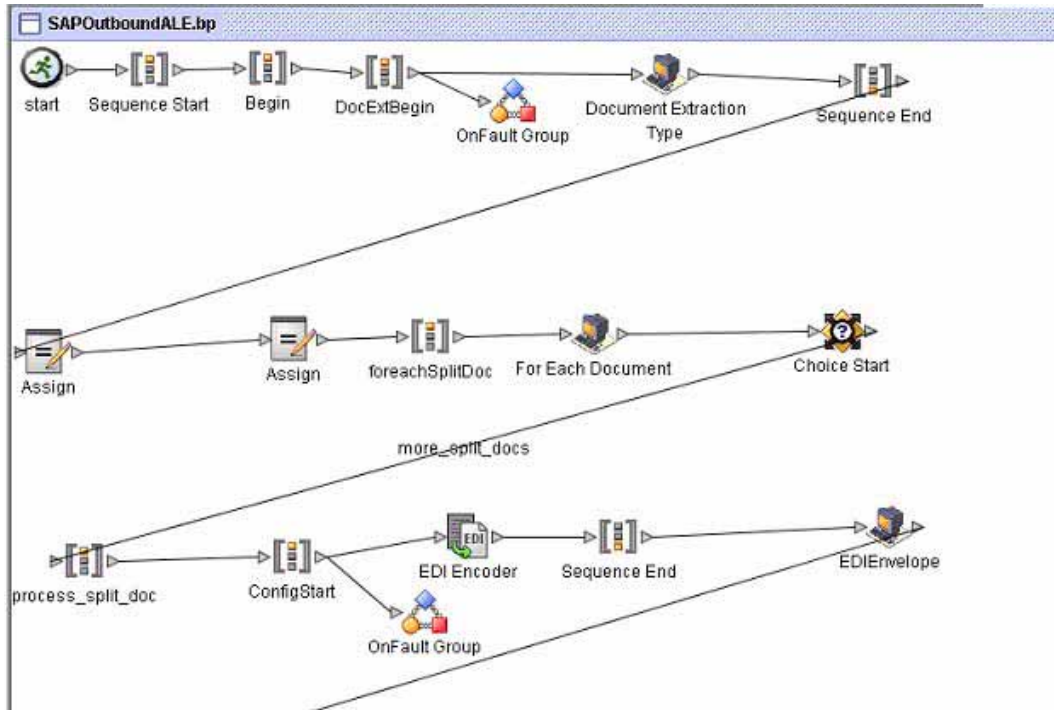
Service Settings	
Service Type	SAP Suite Adapter
Description	Configuration to include in documentation
System Name	ExampleGPMsAPOutboundALE
Group	None
SAP Integration Mode	ALE based IDoc
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	127.0.0.1
Gateway Service	sapgw01
SAP System Number	01
SAP Version	4
Retry Sending IDocs	Yes
Max. Retries (0=unlimited)	5
Retry Sending Interval	10
Client	999
User	sapusername
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	Yes
RFC Server Instances	1
Program ID	SERVER_ALE_TEST
Outbound Process to start	SAPOutboundALE

Specifies the integration mode for interacting with the SAP system.

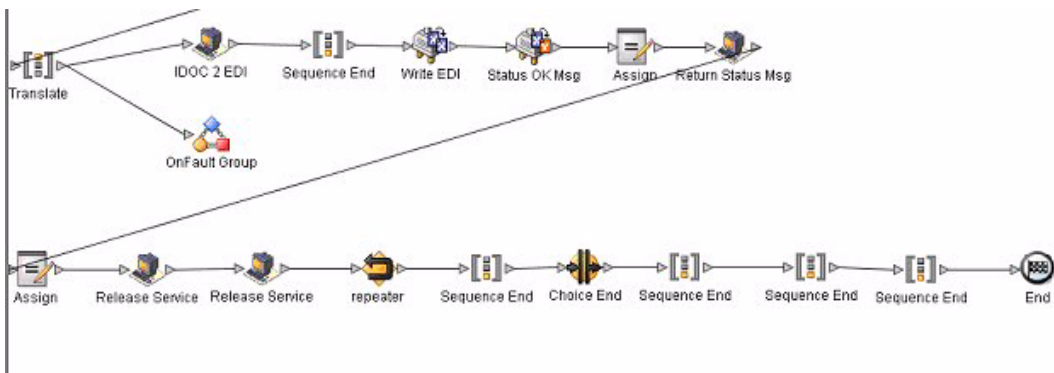
Business process invoked when an ALE IDoc is received from an SAP system

The following example using the GPM illustrates the SAPOutboundALE business process that runs by the SAP Suite adapter for an outbound ALE IDoc. With ALE technology, the IDoc is included in the outbound request and becomes the primary document for the outbound business process that ran. The business process processes the IDoc (translates the IDoc to EDI format and sends it to a trading partner). In addition, the

business process starts a subprocess that uses an instance of the SAP Suite adapter to send a status message back to the SAP system.

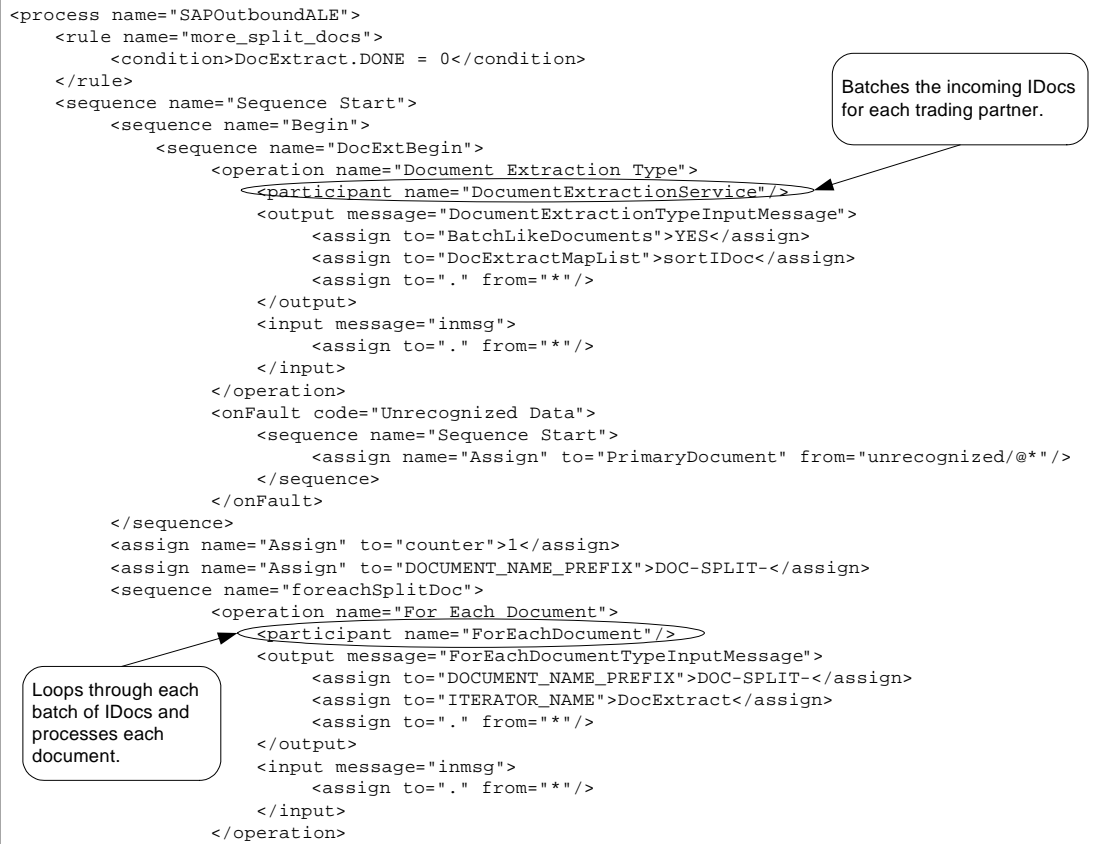


(Screen 1 of 2)



(Screen 2 of 2)

The following example illustrates the same business process using BPML.

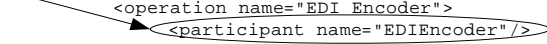


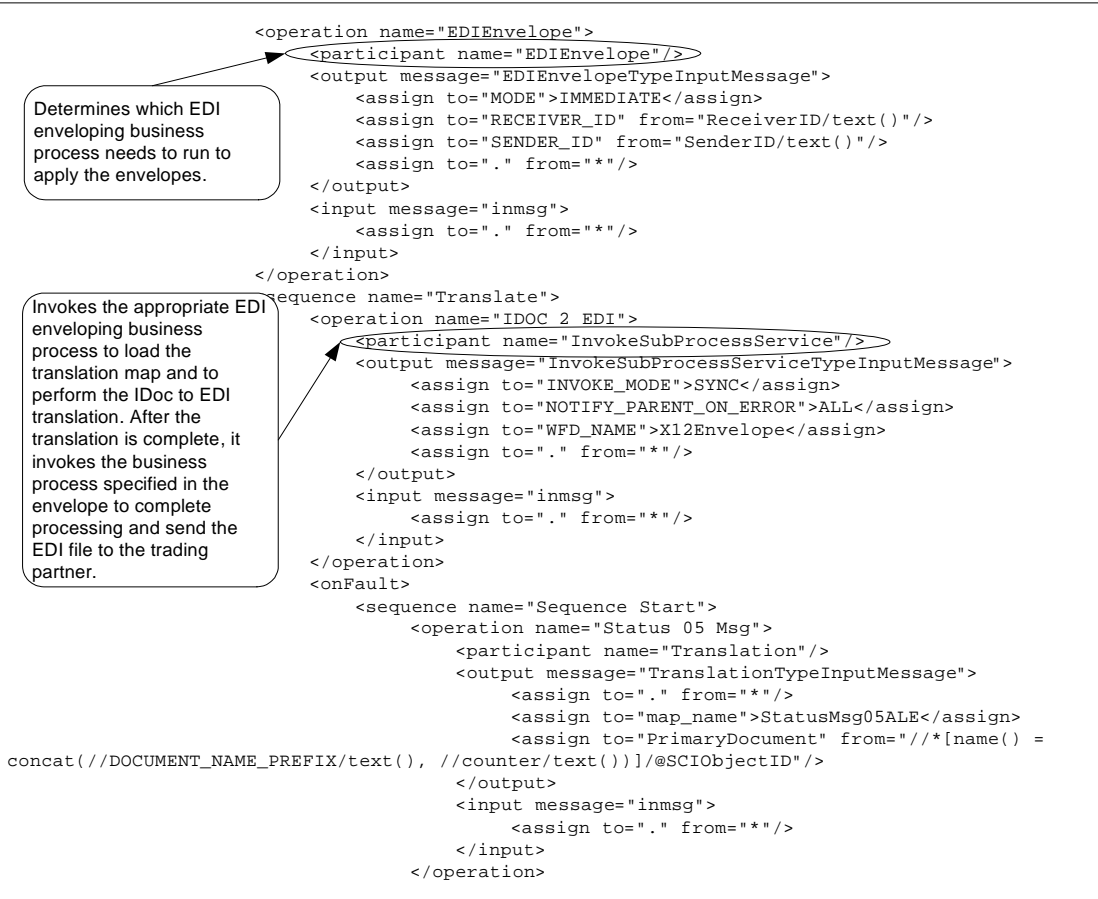
```

<choice name="Choice Start">
  <select>
    <case ref="more_split_docs" activity="process_split_doc"/>
  </select>
  <sequence name="process_split_doc">
    <sequence name="ConfigStart">
      <operation name="EDI Encoder">
        <participant name="EDIEncoder"/>
        <output message="EDIEncoderTypeInputMessage">
          <assign to="AcceptorLookupAlias" from="//*[name() =
concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/AcceptorLookupAlias/text()"/>
          <assign to="ReceiverID" from="//*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text())]/ReceiverID/text()"/>
          <assign to="SenderID" from="//*[name() = concat(//
DOCUMENT_NAME_PREFIX/text(), //counter/text())]/SenderID/text()"/>
          <assign to="." from="*" />
        </output>
        <input message="inmsg">
          <assign to="." from="*" />
        </input>
      </operation>
    </sequence>
    <onFault>
      <sequence name="Sequence Start">
        <operation name="Status 04 Msg">
          <participant name="Translation"/>
          <output message="TranslationTypeInputMessage">
            <assign to="." from="*" />
            <assign to="map_name">StatusMsg04ALE</
assign>
            <assign to="PrimaryDocument" from="//
//*[name() = concat(//DOCUMENT_NAME_PREFIX/text(), //counter/text())]/@SCIOBJECTID"/>
            </output>
            <input message="inmsg">
              <assign to="." from="*" />
            </input>
          </operation>
          <assign name="Assign" to="IsStatusMessage">1</assign>
          <operation name="Return Status Msg">
            <participant name="InvokeSubProcessService"/>
            <output
message="InvokeSubProcessServiceTypeInputMessage">
              <assign to="INVOKE_MODE">ASYNC</assign>
              <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
              <assign to="WFD_NAME">SAPALEDelivery</assign>
              <assign to="." from="*" />
            </output>
            <input message="inmsg">
              <assign to="." from="*" />
            </input>
          </operation>
          <assign name="Assign" to="counter" from="counter + 1"/>
          <repeat name="repeater" ref="foreachSplitDoc"/>
        </sequence>
      </onFault>
    </sequence>
  </choice>

```

Determines which envelope services need to run based on the AcceptorLookupAlias, SenderID, and ReceiverID.



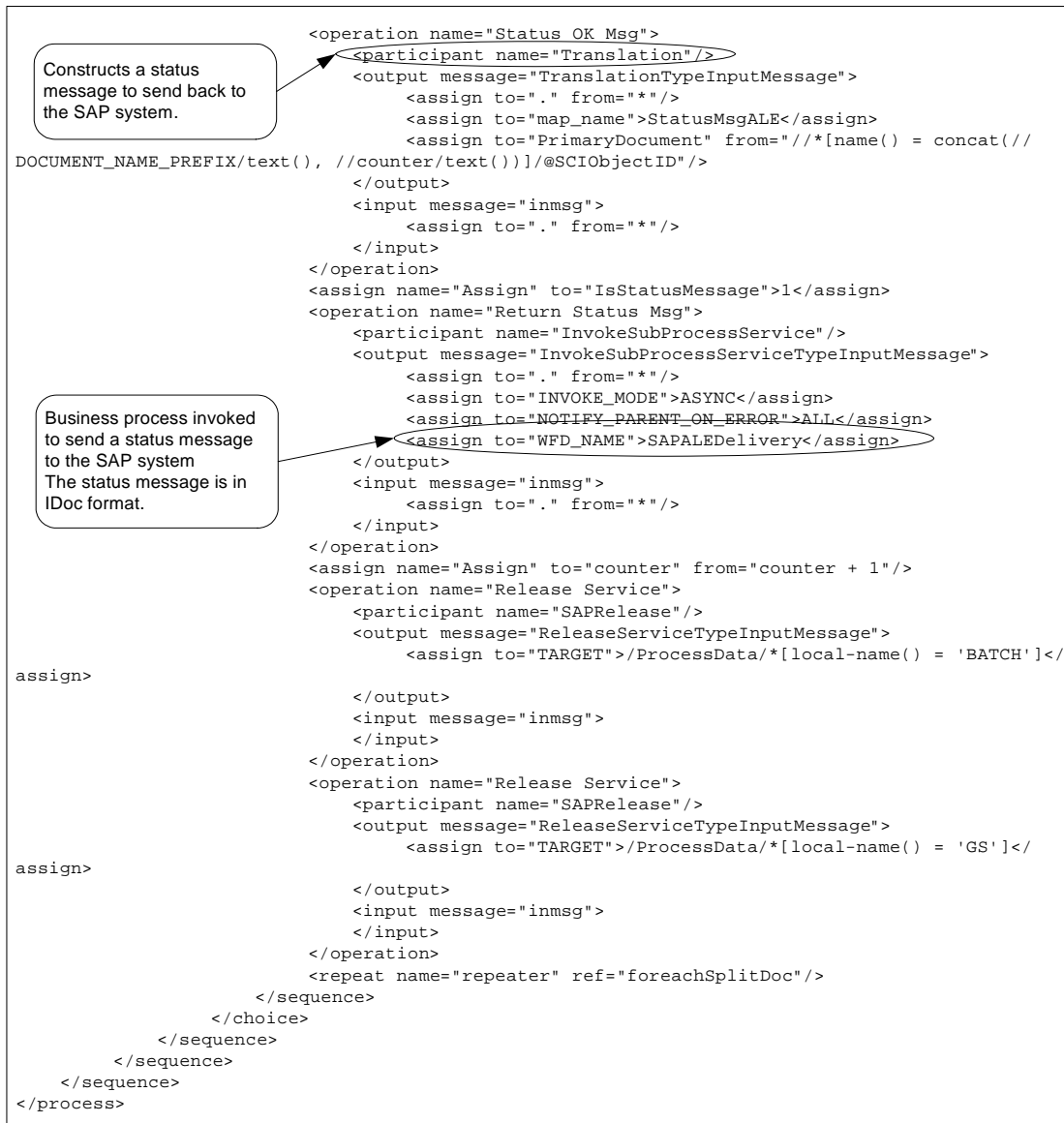


```

    <assign name="Assign" to="IsStatusMessage">1</assign>
    <operation name="Return Status Msg">
      <participant name="InvokeSubProcessService"/>
      <output message="InvokeSubProcessServiceTypeInputMessage">
        <assign to="." from="**"/>
        <assign to="INVOKE_MODE">ASYNC</assign>
        <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
        <assign to="WFD_NAME">SAPALDelivery</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="**"/>
      </input>
    </operation>
    <assign name="Assign" to="counter" from="counter + 1"/>
    <operation name="Release Service">
      <participant name="SAPRelease"/>
      <output message="ReleaseServiceTypeInputMessage">
        <assign to="TARGET">/ProcessData/*[local-name() = 'DOC']</assign>
      </output>
      <input message="inmsg">
        </input>
    </operation>
    <repeat name="repeater" ref="foreachSplitDoc"/>
  </sequence>
</onFault>
</sequence>
<operation name="Write EDI">
  <participant name="FS_WriteEDI"/>
  <output message="FileSystemInputMessage">
    <assign to="assignedFilename" from="concat(//DOCUMENT_NAME_PREFIX/text(), //
counter/text())"/>
    <assign to="assignFilename">true</assign>
    <assign to="extractionFolder">/server1/oracle/ids/</assign>
    <assign to="." from="**"/>
  </output>
  <input message="inmsg">
    <assign to="." from="**"/>
  </input>
</operation>

```

Writes the translated EDI document to the local file system.



The following example illustrates the status message sent to the SAP system upon successful completion of the outbound business process that ran.

```

EDI_DC40 900 46C 2 SYSTAT01 STATUS ALETSTPORTLI 0000001111
SAPI02 LS SITEST 20040408041857

E2STATS001 900 000001000000 EDI_DS40
900000000000006922012004040804185724 Sterling SAPSuite Control
information of EDI subsystem OK
S

E2STATS001 900 000002000000 EDI_DS40
900000000000006922012004040804185706 Sterling SAPSuite

```

TranslationOK

S

```
E2STATS001      900      000003000000  EDI_DS40
900000000000006922012004040804185708      Sterling SAPSuite      Syntax
checkOK
```

S

```
E2STATS001      900      000004000000  EDI_DS40
900000000000006922012004040804185710      Sterling SAPSuite
Interchange handling OK
```

S

An SAP administrator can then view the status messages in the SAP system.

Receiving a Request from SAP and Returning a Synchronous Response Using RFC

This section includes an example SAP Suite adapter configuration and an example business process that runs when a request is received from an SAP system that requires a synchronous response. For example, a trading partner might need a price list for a particular order item before fulfilling the order.

Preconditions:

- ◆ The RFC must exist in the SAP system so the SAP Suite adapter can retrieve the RFC metadata description for it.
- ◆ The RFC must be registered in the RFC server of the SAP Suite adapter so the RFC server can listen for the selected RFC call.

You register an RFC in the SAP Suite adapter configuration instance that receives outbound RFC requests.

Examples:

The following example illustrates a simple custom RFC module Z_TRIGGERSI that starts by the SAP system. This RFC module has two import parameters – PARAM and VALUE – and one export parameter – RES.

```
FUNCTION Z_TRIGGERSI.
*"-----
*" "Locale Interface:
*"  IMPORTING
*"    VALUE(PARAM) TYPE  STRING OPTIONAL
*"    VALUE(VALUE) TYPE  STRING OPTIONAL
*"  EXPORTING
*"    VALUE(RES) TYPE  STRING
*"-----
write 'test'.

ENDFUNCTION.
```

The following example illustrates an SAP Suite adapter configuration used for receiving the RFC request.

► **ExampleGPM SAP Suite Outbound RFC Sync**

Service Settings	
Service Type	SAP Suite Adapter
Description	Documentation example
System Name	ExampleGPM SAP Suite Outbound RFC Sync
Group	None
SAP Integration Mode	Synchronous RFC
SAP system is loadbalanced	No
SAP Application Server	127.0.0.1
Gateway Host	127.0.0.1
Gateway Service	sapgw01
SAP System Number	01
Client	999
User	sapusername
Password	*****
Language	EN
Code Page	1100
Start RFC Server automatically	Yes
RFC Server Instances	1
Program ID	SYNCT
Outbound Process to start	ExampleSAPOutboundRFCSyncResponse
Outbound Encoding	UTF8
Wait for synchronous RFC Outbound response	Yes

Specifies the integration mode for interacting with the SAP system.

Business process invoked when a request for the specified RFC is received from an SAP system

(Screen 1 of 2)

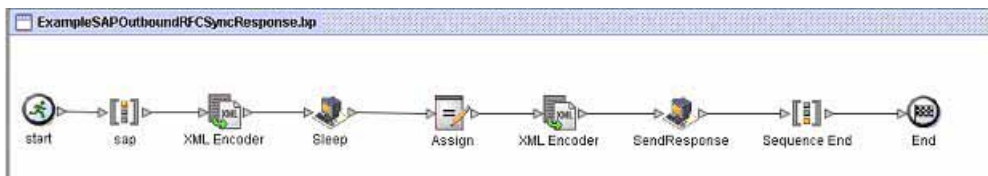
Response timeout (seconds, 0=unlimited)	120	
Delete TIDs automatically	No	
Delete TIDs after (days)	30	
Register Remote Function Calls (BAPI or RFC Mode only)	Enter RFCs offline	Option for registering RFCs
RFC List (comma separated)	Z_TRIGGERSI	The name of the RFC module to invoke
Filter online RFC list by	None provided	
Use hard max. connections limit (Off= soft limit)	Yes	
Max. Connections	4	
Soft Limit Delay Time (seconds)	120	
Connection Check Interval (seconds)	30	
Connection Idle Time (seconds)	240	
Close session after maximum session time	Yes	
Max. Session Time (minutes)	60	
User	admin	

(Screen 2 of 2)

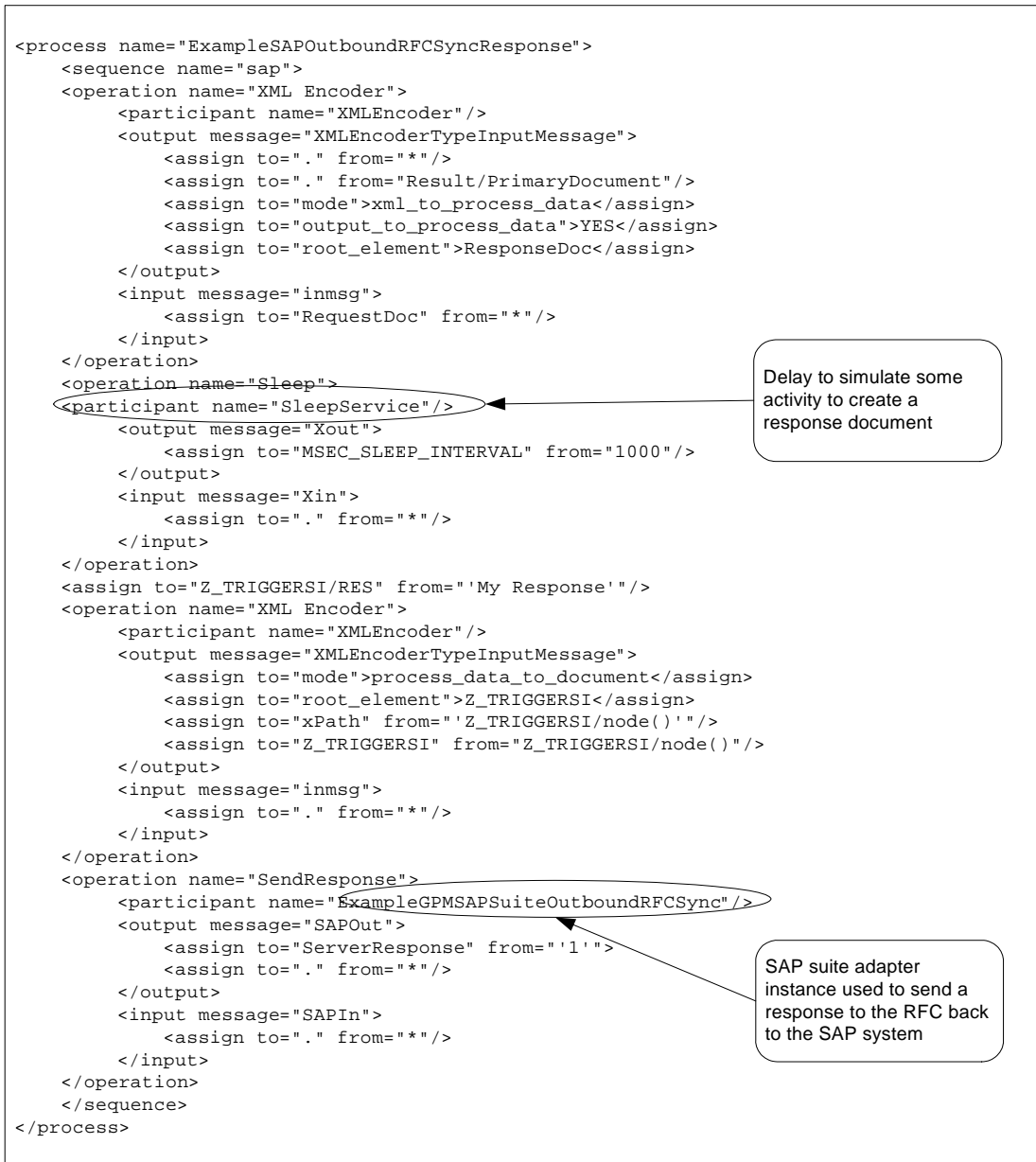
When an outbound RFC is detected by the SAP Suite adapter, the RFC server runs the business process specified on the SAP Suite adapter configuration. The RFC parameters are input to the business process and become the primary document. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<Z_TRIGGERSI>
<PARAM>AAA</PARAM>
<VALUE>BBB</VALUE>
<RES></RES>
</Z_TRIGGERSI>
```

The following example using the GPM illustrates an example business process that starts by the SAP Suite adapter for an outbound RFC request. This business process creates and returns a response to the RFC back to SAP.



The following example illustrates the same business process using BPML.



Note: You must set the ServerResponse parameter to **1** and pass it to the SAP Suite adapter used for sending a response back to SAP. Additionally, you must pass the ServerSessionID and ServerSessionSequenceNumber parameters to the SAP Suite adapter. Both ServerSessionID and ServerSessionSequenceNumber parameters are passed using process data in the previous example. See *Business Process Definition Parameters* on page 1301 for a description of these parameters.

The following example illustrates the response sent back to the SAP system:

```

<Z_TRIGGERSI>
<RES>My Response</RES>

```

</Z_TRIGGERSI>

Connection Retry

When the SAP Suite adapter is started during the application startup, the RFC Server tries to establish a connection to the SAP System configured in the adapter instance. The default behavior of the RFC Server component is to keep trying to establish a connection until it successfully opens a connection.

However, for some permanent login error types, no retry is performed. For example, if the configuration has a wrong SAP user password, then no retry is performed because the error has to be resolved manually (by entering the correct password). Also, no retry is performed if a login fails because of a locked SAP user, which requires the SAP Administrator to manually unlock the user.

Sometimes, the SAP user that is used to login by the SAP Suite adapter is locked during SAP maintenance, and then unlocked after the maintenance. In this case, the *User locked* error must be treated as a temporary error condition and a connection retry is required so that the RFC Server can reconnect automatically after the SAP user is unlocked.

Perform the following steps to enable the connection retry in case of a locked SAP user for a SAP Suite adapter instance:

1. In the *install_dir/properties* directory, add the following line at the end of the *sap.properties* file:

```
SAP.instance_name UnconditionalConnectionRetry = Yes
```


instance_name must be replaced by the name of the SAP Suite adapter instance.
2. Save the file.
3. Restart the application.

Troubleshooting Tips

This section contains troubleshooting tips for using the SAP Suite adapter.

Java Error in the SAP Outbound Business Process

For the *SAPOutboundIDoc.bp*, if the *FS_WriteEDI* service (which is an instance of the File System adapter) is not configured properly to extract data, the Advanced Status column in the Business Process Monitor page displays the following Java error message:

```
java.io.FileNotFoundException
```

In addition, the Status Report column does not provide a report. In this circumstance, the *FS_WriteEDI* service is working as designed. However, the Advance Status Details will show that a file was not found occurs if the *FS_WriteEDI* service is not configured correctly.

SAP XI Adapter

The SAP XI adapter, in conjunction with the GIS Resource Adapter (GIS RA), provides integration capability between Application and SAP® NetWeaver™ through the SAP Exchange Infrastructure (XI) component of SAP NetWeaver.

The following table provides an overview of the SAP XI adapter:

System Name	SAP XI Adapter
Graphical Process Modeler (GPM) category	All Services
Description	The SAP XI adapter connects Application to the SAP XI component of SAP NetWeaver. The SAP XI adapter communicates with SAP XI through the GIS RA (Resource Adapter) supplied with Application and deployed in SAP XI.
Business usage	The SAP XI and GIS RA adapters together are used to send and receive information in the form of messages with XML or binary content between Application and SAP XI. This allows a tight integration between Application and NetWeaver-based SAP solutions.
Usage example	Application translates an incoming EDI ORDER document into an XML-based ORDER message (as defined by SAP) and sends it to the SAP XI system using the SAP XI adapter.
Preconfigured?	No. Both adapters must be installed separately in Application and SAP XI. Both adapters require configuration steps to be performed after installation.
Requires third party files?	The SAP XI adapter requires three library files from your SAP XI installation: <ul style="list-style-type: none">◆ sapj2eeclient.jar◆ exception.jar◆ logging.jar See <i>Installing the SAP XI Library Files</i> on page 1350 for more information. Note: The SAP XI adapter does not require the SAP JCo libraries.
Platform availability	All supported Application platforms and all supported SAP NetWeaver platforms.
Related services	None
Application requirements	An SAP XI account (User ID and password) is required for the SAP XI adapter to communicate with SAP XI. This account must have permissions for rebind/unbind operations in SAP JNDI registry. This can be accomplished by assigning the role SAP_J2EE_ADMIN to the SAP XI user account. Similarly, a Application account (User ID and password) is required for the GIS RA to communicate with Application. The SAP XI adapter must be registered in the Application JNDI registry. The SAP XI adapter and GIS RA must be bound to the SAP XI JNDI Naming service using different JNDI addresses.
Initiates business processes?	The SAP XI adapter initiates a business process in Application when it receives a message from SAP XI. The message contents and any optional attachments are passed to the business process.

Invocation	<p>SAP XI to Application</p> <p>SAP XI triggers an outbound (out of SAP) message. The SAP RA adapter sends the message to Application. The name of the business process to be started for an SAP outbound message is configured in the SAP XI adapter. The SAP XI adapter must be configured as an outbound adapter.</p> <p>Application to SAP XI</p> <p>A Application business process can invoke the SAP XI adapter to send a message to SAP XI. The SAP XI adapter must be configured as an inbound adapter. The message is passed to the SAP XI adapter as the primary document. Attachments may be passed in a XML structure of the process data. When the SAP RA adapter receives the message, it passes it to the SAP AF. Routing information may also be passed in the message.</p>
Business process context considerations	None
Returned status values	<p>Status</p> <ul style="list-style-type: none"> ◆ Success – SAP XI ended successfully ◆ Error – Various advanced status values plus process data content <p>Advanced Status</p> <p>The SAP XI adapter returns one of the following messages if an error occurs during message processing:</p> <ul style="list-style-type: none"> ◆ GA_INITIALIZATION_FAILED – Error during SAP XI adapter, including communication failures such as “bind failed”. ◆ GA_SYNC_SEND_FAILED – Error using the sendSync method, including communication failures. ◆ GA_FAILURE – Unspecified error. Information is contained in the status report or the sapxi.log file.
Restrictions	None
Persistence level	The SAP XI adapter bootstraps a business process in inbound mode. The persistence level is determined by the persistence settings of the bootstrapped business process.
Testing considerations	<p>SAP XI Adapter</p> <p>During testing, set the log level for the sapxi logger to ALL in the <i>install_dir/properties/log.properties</i> file.</p> <p>GIS RA</p> <p>During testing, turn on Debug mode, in SAP XI, for the GIS RA (called GIS adapter in SAP XI) to save all trace messages. The SAP XI trace files are saved in the <i>/usr/sap/SysNum/DVEBMGS00/j2ee/cluster/servernum/log</i> directory with the file name <i>defaultTrace.num.trc</i>.</p>

How the SAP XI Adapter Works

The SAP XI adapter integrates Application with SAP NetWeaver through the XI Adapter Framework (AF) of the SAP XI 3.0 component of SAP NetWeaver. SAP recommends SAP XI as the component for exchanging data with external applications on either SAP or non-SAP systems. The XI AF, specifically designed for adapter development and support, runs on the SAP J2EE Application Server (SAP WebAS) and inherits many of the advantages of the SAP WebAS platform, including:

Scalability
Reliability
Persistence layer
Transport and security mechanisms

The integration between Application and SAP NetWeaver is actually accomplished by two adapters:

SAP XI adapter – A stateful adapter which is deployed and runs in Application.

GIS RA – A JCA-compliant resource adapter that is deployed and runs in the SAP XI Adapter Framework (AF) of SAP NetWeaver.

The SAP XI adapter allows you to:

Send messages from Application to SAP XI and receive synchronous responses from SAP XI. (Configured as an inbound adapter.)

Receive messages from SAP XI, initiate business processes in Application, and, optionally, send synchronous responses back to SAP XI. (Configured as an outbound adapter.)

The GIS RA allows you to:

Send messages from SAP XI to the Application SAP XI adapter and receive synchronous responses.

Receive messages from the Application SAP XI adapter, pass them to the SAP XI AF, and send back synchronous responses.

SAP XI adapter configurations create a 1:1 connectivity to related GIS RA channels. SAP channels are similar to adapter configurations in Application. The connectivity depends on whether you are sending or receiving messages:

1. An inbound SAP XI adapter configuration in Application must be configured to connect to an outbound GIS RA Sender channel in SAP XI. A Sender channel sends messages into the SAP XI AF.
2. An outbound SAP XI adapter configuration in Application must be configured to connect to an inbound GIS RA Receiver channel in SAP XI. A Receiver channel receives messages from the SAP XI AF.

SAP XI Messages

Messages consist of a message payload and, optionally, one or more attachments. The content of the message may be XML or binary. If the content is binary, the message cannot be routed dynamically within SAP XI. The content of the attachments is arbitrary. Attachments are passed to the receiving partner in SAP XI without being opened or parsed. Messages can be sent in synchronous mode only.

Inbound Messages

Messages sent from Application to SAP XI are called inbound messages. The message payload is passed as a primary document in the root of the process data. One or more attachments can also be sent with the message in an attachments structure.

Input Message

The BPML format for the message is:

```
<PrimaryDocument SCIObjectID=id
```

The optional attachments structure includes a list of attachment nodes:

```
<attachments>
  attachment_1
  ...
  attachment_n
```

This allows for multiple attachments. The attachment numbers increment by one for each attachment and *n* is the number of the final attachment.

The structure for a single attachment is:

Each attachment consists of a document node and an optional body property part. The document node contains a primary document with the attachment contents. You can specify an optional index attribute to define the order of attachments in the message. If the index attribute is omitted, attachments are processed in the specified order.

The structure for a basic document node without optional index attributes or body properties is:

```

  n
  PrimaryDocument
  n
```

The structure for a document node with optional index attributes is:

```

  n
  PrimaryDocument
  <properties index=n>
  BodyProperties
  n
```

Body properties describe details of a payload message or attachment message. The structure for body properties is:

```

  name
  description
  version
  schema
  encoding
  type
  or
```

Response Message

The response message consists of the response payload in the primary document and the SAPXIResponse structure which contains header and body details of the payload and an optional attachments part.

```
PrimaryDocument
```

HeaderDetails

BodyProperties

Attachments **(optional)**

Header Details

```
<Header>
  <ToService>      </ToService>
  <FromService>    </FromService>
  <ToParty>        </ToParty>
  <FromParty>      </FromParty>
  <MessageID>      </MessageID>
  <Action>         </Action>
  <ActionNS>      </ActionNS>
</Header>
```

Outbound Messages

Messages sent from SAP XI to Application are called outbound messages. For outbound messages, a business process is specified in an SAP XI adapter configuration. This business process is bootstrapped when a message is received from SAP XI. The Output Message is returned in the process data of the business process.

Output Message

The output message consists of the contents of the primary document, an SAP XI outbound structure and an originator ID.

The BPML format for the message is:

```
SAP XI Outbound Structure
  id
```

The SAP XI Outbound structure is:

BodyProperties

Attachments **Optional**

Response Message

The bootstrapped business process can return a synchronous response to the sender in the same call passing the Originator ID to the outbound adapter configuration. To receive a synchronous response:

1. **Wait for synchronous response** must be selected. See *Creating or Setting Up an SAP XI Adapter Configuration in the Admin Console* on page 1351.
2. **XICallType** must be set to **SYNCHRONOUS_RESPONSE**. See *Setting Up the SAP XI Adapter in the GPM* on page 1352.

- ◆ install3rdParty.cmd sapxi 30 -j \sapj2eeclient.jar
- ◆ install3rdParty.cmd sapxi 30 -j \exception.jar
- ◆ install3rdParty.cmd sapxi 30 -j \logging.jar

5. Start Application.

Configuring the SAP XI Adapter

You must specify field settings in Application, using the Admin Console, and in the GPM.

Creating or Setting Up an SAP XI Adapter Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the SAP XI adapter, or to set up the configuration provided with Application. Some fields are available in both the Admin Console and in the GPM. For the fields that are available in both, the GPM field name is shown in parentheses in the table below.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select one of the following options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Managing Services and Adapters</i>.</p>
Service Type (ServiceType)	<p>Identifies the SAP XI adapter as an inbound (to SAP XI) or outbound (from SAP XI) adapter. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ SAP XI Inbound – (Default) Message flow from Application to SAP XI. ◆ SAP XI Outbound – Message flow from SAP XI to Application.
Business Process Name (BootstrapBP)	Select a business process to bootstrap when a message is received from SAP XI. Only displays if Service Type is set to SAP XI Outbound . Required. Valid values: List of business processes in Application.
Wait for synchronous response (SyncWait)	If checked, the incoming synchronous call is delayed until a response from the initiated business process is returned. Only displays if Service Type is set to SAP XI Outbound .
Synchronous response timeout [s] (SyncWaitTimeout)	Number of seconds to wait for a response from the initiated business process before issuing a Timeout Exception. Only displays if Service Type is set to SAP XI Outbound . Required. Valid value is any number. Default is 60.

Field	Description
SAP XI JNDI Host Name and Port (Host:Port) (JNDIRemoteAddress)	Host and JNDI port of the remote SAP XI JNDI Server. Required. Syntax is <code>Host : Port</code>
SAP XI JNDI Key (JNDIRemoteKey)	JNDI key used to look up the remote RMI server in SAP XI. Required. Use a unique ID for every adapter configuration. Default is <code>com.sterlingcommerce.woodstock.services.sapxira</code>
SAP XI JNDI Context (RemoteJNDIContext)	Initial JNDI context of the remote SAP XI JNDI server. Required. This is defined by the SAP JNDI implementation. Default is <code>com.sap.engine.services.jndi.InitialContextFactoryImpl</code>
SAP XI JNDI User Name (RemoteUser)	User of the remote SAP XI JNDI Server. The SAP XI user must have permissions for JNDI bind and unbind operations (for example, role <code>SAP_J2EE_ADMIN</code>). Required.
SAP XI JNDI Password (RemotePassword)	Password of the remote SAP XI JNDI Server. Required.
Registry Type for Local Server (LocalRegistryType)	The JNDI Registry to use for the SAP XI Adapter. Required. Select the following option: <ul style="list-style-type: none"> ◆ Using External Registry – Use an external JNDI server. All JNDI connection parameters must be entered.
GIS JNDI Host Name and Port (Host:Port) (JNDILocalAddress)	Host and JNDI port of the local JNDI Server. This is the external JNDI Server used by the SAP XI adapter. Required. Only displays if Registry Type for Local Server is set to Using External Registry . Syntax is <code>Host : Port</code>
GIS JNDI Key (JNDILocalKey)	JNDI key used to look up the local RMI Server used by the SAP XI adapter. Use a unique ID for every adapter configuration. Required. Default is <code>com.sterlingcommerce.woodstock.services.sapxiga</code>
GIS JNDI Context (LocalJNDIContext)	Initial JNDI Context of the local JNDI Server. This is the JNDI Server used by the SAP XI adapter. Required. Default is <code>org.jnp.interface.NamingContextFactory</code> . Change default to the following value: <code>com.sap.engine.services.jndi.InitialContextFactoryImpl</code>
GIS JNDI User Name (LocalUser)	User of the Local JNDI Server. Required. Only displays if Registry Type for Local Server is set to Using External Registry .
GIS JNDI Password (LocalPassword)	Password of the Local JNDI Server. Required. Only displays if Registry Type for Local Server is set to Using External Registry .

Setting Up the SAP XI Adapter in the GPM

Use the field definitions in the following table to set up the adapter configuration in the GPM:

Field	Description
XICallType	Specifies the message handling mode. Required. Valid value: <ul style="list-style-type: none"> ◆ SYNCHRONOUS – Use for an inbound adapter ◆ SYNCHRONOUS_RESPONSE – Use to pass back a synchronous response with an outbound adapter
AttachmentRoot	If set, the parameter refers to an XML structure in process data containing the attachment documents of the message. If omitted, the root "attachments" is used. Sending attachments in a message is optional. Optional. Valid value is a string that defines the Root Tag. Default is attachments.
ToParty	ToParty field for the SAP XI Message envelope. May be required or optional depending on the SAP XI scenario.
FromParty	FromParty field for the SAP XI Message Envelope. May be required or optional depending on the SAP XI scenario. String.
ToService	ToService field for the SAP XI Message Envelope. Required. String.
FromService	FromService field for the SAP XI Message Envelope. Required. String.
Action	Action field for the SAP XI Message Envelope. Required. String.
ActionNS	Action Name Space field for the XI Message Envelope. Required. String.
MessageID	Unique Message ID on Application side for the XI Message Envelope. Optional. If omitted, a GUID is generated automatically. Default is GUID.
MessageDescription	Message Description field for the XI Message Envelope. Optional. String.
BodyName	Body name of the SAP XI Message. Optional. String.
BodyDescription	Body description of the SAP XI Message. Optional. String.
BodyVersion	Body version of the SAP XI Message. Optional. String.
BodySchema	XML only. Body name of the SAP XI message. String. Recommended.
BodyEncoding	Character set used for the SAP XI message. String. Default is UTF-8. Recommended.
BodyContentType	Body content type of the SAP XI message. String. Optional.
IsXMLContent	Specifies whether the SAP XI message content is in XML format. Boolean string. Recommended.
OriginatorID	Outbound only. Uniquely identifies the outbound session waiting for the synchronous response. The OriginatorID is generated automatically by the outbound adapter in the process data and must be passed back in the response call. Required.

Global Properties for the SAP XI Adapter

There are optional global properties that can be configured for the SAP XI adapter. See *sapxi.properties* in the Application Documentation Library for information.

Installing and Configuring the GIS Resource Adapter within SAP XI

The SAP XI Resource Adapter (GIS RA) is packaged in an SAP-specific installation called a Software Delivery Archive (SDA). This single SDA file (`gis_xi_adapter_version.sda`) contains all the files necessary to deploy the GIS RA in SAP NetWeaver using the Software Deployment Manager (SDM) tool.

Installing and configuring the GIS RA in SAP XI requires the following procedures:

- Copying the SDA to SAP NetWeaver on page 1354*
- Deploying the SDA in SAP NetWeaver on page 1354*
- Adding the RA to the System Landscape Directory on page 1354*
- Creating a Repository Namespace on page 1355*
- Creating an Adapter Object on page 1355*

Copying the SDA to SAP NetWeaver

1. Locate the following SDA file on the machine where Application is installed:
`install_dir/packages/gis_xi_adapter_version.sda`
2. Copy the SDA file to the machine where the Adapter Framework (AF) of the SAP XI system is running.

Deploying the SDA in SAP NetWeaver

1. Open the SDM GUI in SAP NetWeaver.
2. Click the **Deployment** tab.
3. On the *Step 1 Choose SCA's/SDA's to be deployed* page, click +.
4. Add the GIS RA into the list of adapters to prepare for deployment.
5. Click **Next**.
6. On the *Step 2 Show deployment Proposal for SCA's/SDA's to be deployed* page, click **Next**.
7. On the *Step 3 SDM is ready to deploy, Start deployment* page, click **Start Deployment**.
8. Wait for the deployment to complete, this may take some time depending on the performance of your SAP XI system.
9. Click **Commit**.
10. Close the SDM.

Adding the RA to the System Landscape Directory

After the SDA is deployed, the GIS RA must be added to the System Landscape Directory (SLD). You must define the RA as a software unit in SAP NetWeaver. To add the GIS RA to the SLD, perform the following steps:

1. Open the SLD in SAP NetWeaver.
2. Select **Home > Products** to display the Software Catalog.
3. Select the **Products** tab.

4. Under **Software Type**, select **Products**.
5. Click **New Product**.
6. On the Define Product screen, complete the following fields:
 - ◆ Vendor
 - ◆ Name
 - ◆ Version
7. Click **Create**.
8. Under **Software Type**, select **Software Components**.
9. Click **New Component**.
10. On the Define Software Component screen, complete the following fields:
 - ◆ Vendor
 - ◆ Name
 - ◆ Version
11. Click **Create**.

Creating a Repository Namespace

After creating the product and software component, you must create a namespace in the integration repository using the SAP Integration Builder. To create a repository namespace, perform the following steps:

1. From the Design screen of the Integration Builder, select the Product Version that you created in the step above from the tree on the left side of the screen. If it does not appear, select **Environment > Clear SLD Data Cache** to refresh the SLD cache.
2. Right-click the product version and click **New Namespace** to access the Namespace editor.
3. In the **Namespaces** table, create a new namespace named **http://stercomm.com/adapter/gis**.

Creating an Adapter Object

After creating the repository namespace, you must create an adapter object that defines an adapter type for the GIS RA. An adapter object is comprised of adapter metadata that defines configuration data for the adapter type and one or more communication channels. The adapter metadata defines the portion of a communication channel that is unique to the adapter type. A communication channel defines the rules for handling messages during inbound or outbound processing.

Setting Up the Adapter Metadata

To set up the adapter metadata, perform the following steps:

1. From the Objects tab in the Integration Builder, locate the namespace **http://stercomm.com/adapter/gis** inside your product version.
2. Click the arrow adjacent to the namespace to display the subheadings.
3. Click the arrow adjacent to **Adapter Objects**.

4. Right-click **Adapter Metadata** and click **New**.
5. In the Name box, type **GIS**.
6. Click **Create**.
7. From the Adapter Metadata editor, click **Upload XML Description from File** in the top speedbutton line.
8. Provide the complete path to the GISAdapter.xml file provided with Application.
9. Select **Adapter Metadata > Save** to save and activate the adapter metadata.


Creating Communication Channel Templates

Communication channels define rules for handling messages during inbound or outbound processing. Depending on the direction of message processing, you require either a sender or a receiver channel. Therefore, you must create two communication channel templates, one for sending and one for receiving.

To create the templates, perform the following steps:

1. From the Objects tab in the Integration Builder, locate the namespace `http://stercomm.com/adapter/gis`.
2. Click the arrow adjacent to the namespace to display the subheadings.
3. Click the arrow adjacent to **Adapter Objects**.
4. Right-click **Communication Channel Templates** and click **New**.
5. From the Parameters tab, specify the following settings:

Field	Description
Name	Specifies a name for the channel template. Note: The Namespace and Software Component Version fields will be pre-filled.
Description	Provides a brief description of the template.
Adapter Type	Specifies the adapter type. Select GIS .
Sender/Receiver	Defines the channel as either a sender or receiver channel. Select one of the following radio buttons: <ul style="list-style-type: none"> ◆ Sender – Used for inbound (to SAP) message processing. ◆ Receiver – Used for outbound (from SAP) message processing.
Transport Protocol	Specifies the transport protocol. Select JCA .
Message Protocol	Specifies the message protocol. Select JCA .

6. Click the Select Attributes button .
7. Add the following attributes by highlighting each field name in the **Available Fields** column and clicking **Add** to move it to the **Show These fields in This Order** column:
 - ◆ SAP XI JNDI Host Name and Port (Host:Port)
 - ◆ SAP XI JNDI Key
 - ◆ SAP XI JNDI Context

- ◆ SAP XI JNDI User Name
- ◆ SAP XI JNDI Password
- ◆ GIS JNDI Host Name and Port (Host:Port)
- ◆ GIS JNDI Key
- ◆ GIS JNDI Context
- ◆ GIS JNDI User Name
- ◆ GIS JNDI Password
- ◆ Quality of Service
- ◆ Channel Address Mode

8. After all fields have been moved to the **Show These fields in This Order** column, click **Apply**.
9. Configure the channel attributes. Specify the following field settings:

Field	Description
SAP XI JNDI Connection Parameters	
SAP XI JNDI Host Name and Port (Host:Port)	SAP XI JNDI Host Name and Port (Host:Port). Required. Valid values: <ul style="list-style-type: none"> ◆ Host:Port ◆ IP:Port
SAP XI JNDI Key	SAP XI JNDI Key. Required. Default is com.sterlingcommerce.woodstock.services.sapxira
SAP XI JNDI Context	SAP XI JNDI Context. Required. Default is com.sap.engine.services.jndi.InitialContextFactoryImpl
SAP XI JNDI User Name	SAP XI JNDI User Name. The SAP XI User must have permissions for JNDI bind and unbind operations (e.g. role SAP_J2EE_ADMIN). Required.
SAP XI JNDI Password	SAP XI JNDI Password. Required.
GIS JNDI Connection Parameters	
GIS JNDI Host Name and Port (Host:Port)	Application JNDI Host Name and Port (Host:Port). Required.
GIS JNDI Key	Application JNDI Key. Required. Default is com.sterlingcommerce.woodstock.services.sapxiga
GIS JNDI Context	Application JNDI Context. Required. Default is org.jnp.interfaces.NamingContextFactory Change default to the following value: com.sap.engine.services.jndi.InitialContextFactoryImpl
GIS JNDI User Name	Application JNDI User Name. Required.

Field	Description
GIS JNDI Password	Application JNDI Password. Required.
Message Parameters	
Quality of Service	Quality of Service. Required. Select Best Effort .
Channel Address Mode	Channel Address Mode. Required. Valid values: <ul style="list-style-type: none"> ◆ CPA = Channel ◆ MSG = Message

10. Repeat the above steps to create the second communication channel. If you created a sender channel, create a receiver channel.

The GIS RA is now set up as a standard SAP XI resource adapter and can be used with the communication channel templates in SAP integration scenarios. See the SAP documentation for more information.

Business Process Examples

The following examples illustrate how the SAP XI adapter can be used in a business process for both inbound and outbound message processing:

Inbound Message Processing (Application to SAP XI):

In this example, the input is the primary document containing the message. The sapxiinbound process sends a message to SAP XI. The message consists of the content from the primary document, the header from workflow parameters, and two attachments.

Process data contents before call

```
<ProcessData>
  <PrimaryDocument SCIObjectID="serverNode:169ca65:109f0bef19c:2f04" />
  <att1>
    <message>This is my attachment 1 message</message>
  </att1>
  <doc1>
    <PrimaryDocument SCIObjectID="          :169ca65:109f0bef19c:2f25" />
  </doc1>
  <att2>
    <message>This is my attachment 2 message</message>
  </att2>
  <doc2>
    <PrimaryDocument SCIObjectID="          :169ca65:109f0bef19c:2f47" />
  </doc2>
  <attachments>
    <attachment_1>
      <document index="1">
        <PrimaryDocument SCIObjectID="          :169ca65:109f0bef19c:2f25" />
      </document>
      <properties>
        <BodyName>BodyName1</BodyName>
        <BodyDescription>BodyDescription1</BodyDescription>
      </properties>
    </attachment_1>
  </attachments>
</ProcessData>
```

```

    <BodyVersion>BodyVersion1</BodyVersion>
    <BodySchema>BodySchema1</BodySchema>
    <BodyEncoding>UTF-8</BodyEncoding>
    <BodyContentType>text/xml</BodyContentType>
    <IsXMLContent>>true</IsXMLContent>
  </properties>
</attachment_1>
<attachment_2>
  <document index="2">
    <PrimaryDocument SCIOBJECTID="          :169ca65:109f0bef19c:2f47"/>
  </document>
  <properties>
    <BodyName>BodyName2</BodyName>
    <BodyDescription>BodyDescription2</BodyDescription>
    <BodyVersion>BodyVersion2</BodyVersion>
    <BodySchema>BodySchema2</BodySchema>
    <BodyEncoding>UTF-8</BodyEncoding>
    <BodyContentType>text/xml</BodyContentType>
    <IsXMLContent>>true</IsXMLContent>
  </properties>
</attachment_2>
</attachments>
</ProcessData>

```

Message To Service

```

<SAPXIOut>
  <LocalPassword>*****</LocalPassword>
  <RemotePassword>*****</RemotePassword>
  <JNDILocalAddress>          :50004</JNDILocalAddress>
<RemoteJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl</RemoteJNDIContext>
  <ServiceType>Inbound</ServiceType>

<LocalJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl</LocalJNDIContext>
  <JNDIRemoteAddress>          :50004</JNDIRemoteAddress>
  <LocalRegistryType>EXTERNAL_REG</LocalRegistryType>
  <RemoteUser>          </RemoteUser>
  <LocalUser>          </LocalUser>
  <JNDILocalKey>com.sterlingcommerce.woodstock.services.sapxiga</JNDILocalKey>
  <XMLVersion>1.0</XMLVersion>
  <JNDIRemoteKey>com.sterlingcommerce.woodstock.services.sapxira</JNDIRemoteKey>
  <XICallType>SYNCHRONOUS</XICallType>
  <AttachmentRoot>attachments</AttachmentRoot>
  <ToParty/>
  <FromParty/>
  <ToService>I03_OutboundToGIS</ToService>
  <FromService>I03_InboundFromGIS</FromService>
  <Action>CarstensMessageInterfaceSync</Action>
  <ActionNS>http://stercomm.com/adapter/gis</ActionNS>
  <MessageID>MSG_DV_001</MessageID>
  <MessageDescription>This is a test message description</MessageDescription>
  <BodyName>BodyName</BodyName>
  <BodyDescription>BodyDescription</BodyDescription>
  <BodyVersion>BodyVersion</BodyVersion>
  <BodySchema>BodySchema</BodySchema>

```

```

<BodyEncoding>UTF-8</BodyEncoding>
<BodyContentType>text/xml</BodyContentType>
<IsXMLContent>>true</IsXMLContent>
</SAPXIOut>

```

BPML Business Process (Inbound)

```

<process name="callXI">
  <sequence name="sap">
    <!-- create two attachments -->
    <assign to="att1/message" from="'This is my attachment 1 message'"/></assign>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">process_data_to_document</assign>
        <assign to="root_element">att1</assign>
        <assign to="XPath" from="'att1/node()'"></assign>
        <assign to="att1" from="att1/node()"/></assign>
      </output>
      <input message="inmsg">
        <assign to="doc1" from="*"></assign>
      </input>
    </operation>

    <assign to="att2/message" from="'This is my attachment 2 message'"/></assign>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">process_data_to_document</assign>
        <assign to="root_element">att2</assign>
        <assign to="XPath" from="'att2/node()'"></assign>
        <assign to="att2" from="att2/node()"/></assign>
      </output>
      <input message="inmsg">
        <assign to="doc2" from="*"></assign>
      </input>
    </operation>

    <!--Create attachment structure with two attachments -->
    <assign to="attachments/attachment_1/document" from="doc1/node()"/></assign>
    <assign to="attachments/attachment_1/document/@index" from="'1'"/></assign>
    <assign to="attachments/attachment_1/properties/BodyName"
from="'BodyName1'"/></assign>
    <assign to="attachments/attachment_1/properties/BodyDescription"
from="'BodyDescription1'"/></assign>
    <assign to="attachments/attachment_1/properties/BodyVersion"
from="'BodyVersion1'"/></assign>
    <assign to="attachments/attachment_1/properties/BodySchema"
from="'BodySchema1'"/></assign>
    <assign to="attachments/attachment_1/properties/BodyEncoding"
from="'UTF-8'"/></assign>
    <assign to="attachments/attachment_1/properties/BodyContentType"
from="'text/xml'"/></assign>
    <assign to="attachments/attachment_1/properties/IsXMLContent"
from="'true'"/></assign>
    <assign to="attachments/attachment_2/document" from="doc2/node()"/></assign>
    <assign to="attachments/attachment_2/document/@index" from="'2'"/></assign>

```



```

    <assign to="attachments/attachment_2/properties/BodyName"
from="'BodyName2' "></assign>
    <assign to="attachments/attachment_2/properties/BodyDescription"
from="'BodyDescription2' "></assign>
    <assign to="attachments/attachment_2/properties/BodyVersion"
from="'BodyVersion2' "></assign>
    <assign to="attachments/attachment_2/properties/BodySchema"
from="'BodySchema2' "></assign>
    <assign to="attachments/attachment_2/properties/BodyEncoding"
from="'UTF-8' "></assign>
    <assign to="attachments/attachment_2/properties/BodyContentType"
from="'text/xml' "></assign>
    <assign to="attachments/attachment_2/properties/IsXMLContent"
from="'true' "></assign>

<!-- Now send message to SAP XI system -->
<operation name="callsapxi">
  <participant name="SendToXI"/>
  <output message="SAPXIOut">
    <assign to="XICallType" from="'SYNCHRONOUS' "></assign>
    <assign to="AttachmentRoot" from="'attachments' "></assign>
    <assign to="ToParty" from="' ' "></assign>
    <assign to="FromParty" from="' ' "></assign>
    <assign to="ToService" from="'I03_OutboundToGIS' "></assign>
    <assign to="FromService" from="'I03_InboundFromGIS' "></assign>
    <assign to="Action" from="'CarstensMessageInterfaceSync' "></assign>
    <assign to="ActionNS" from="'http://stercomm.com/adapter/gis' "></assign>
    <assign to="MessageID" from="'MSG_DV_001' "></assign>
    <assign to="MessageDescription" from="'This is a test message
description' "></assign>
    <assign to="BodyName" from="'BodyName' "></assign>
    <assign to="BodyDescription" from="'BodyDescription' "></assign>
    <assign to="BodyVersion" from="'BodyVersion' "></assign>
    <assign to="BodySchema" from="'BodySchema' "></assign>
    <assign to="BodyEncoding" from="'UTF-8' "></assign>
    <assign to="BodyContentType" from="'text/xml' "></assign>
    <assign to="IsXMLContent" from="'true' "></assign>
    <assign to="." from="*" "></assign>
  </output>
  <input message="SAPXIIn">
    <assign to="." from="*" "></assign>
  </input>
</operation>
</sequence>
</process>

```

Outbound Message Processing (SAP XI to Application):

In this example, the input is an SAP XI message with two attachments and the output is a response message with one attachment. The `sapxiout_response` process receives a message from SAP XI. The message consists of the message content and two attachments. Within the business process, a response message is created and passed back synchronously to the SAP XI system in the same call.

Process data directly after the business process is bootstrapped:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```

<ProcessData>
  <PrimaryDocument SCIObjectID="          :169ca65:109de7de0fd:-5612"/>
  <SAPXIOutbound>
    <Header>
      <ToService>I03_OutboundToGIS</ToService>
      <FromService>I03_InboundFromGIS</FromService>
      <FromParty/>
      <MessageID>c41268c0-af56-11da-b4bc-0003ba783746</MessageID>
      <ToParty/>
    </Header>
    <attachments>
      <attachment_1>
        <document index="1">
          <PrimaryDocument SCIObjectID="          :169ca65:109de7de0fd:-560c"/>
        </document>
        <properties index="1">
          <BodyVersion>undef</BodyVersion>
          <BodyEncoding>UTF-8</BodyEncoding>
          <BodyName>BodyName1</BodyName>
          <IsXMLContent>>false</IsXMLContent>
          <BodyDescription>BodyDescription1</BodyDescription>
          <BodyContentType>text/xml</BodyContentType>
        </properties>
      </attachment_1>
      <attachment_2>
        <properties index="2">
          <BodyDescription>BodyDescription2</BodyDescription>
          <BodyContentType>text/xml</BodyContentType>
          <BodyEncoding>UTF-8</BodyEncoding>
          <BodyName>BodyName2</BodyName>
          <IsXMLContent>>false</IsXMLContent>
          <BodyVersion>undef</BodyVersion>
        </properties>
        <document index="2">
          <PrimaryDocument SCIObjectID="          :169ca65:109de7de0fd:-560a"/>
        </document>
      </attachment_2>
    </attachments>
    <properties>
      <BodyVersion/>
      <BodyName>BodyName</BodyName>
      <BodyDescription>BodyDescription</BodyDescription>
      <IsXMLContent>>true</IsXMLContent>
      <BodyContentType>text/xml</BodyContentType>
      <BodyEncoding>utf-8</BodyEncoding>
      <BodySchema/>
    </properties>
  </SAPXIOutbound>
  <OriginatorID>          .169ca65.109de7de0fd.F5647</OriginatorID>
</ProcessData>

```

Additional Process data before response call:

...following process data is added by the business process to prepare the response...

```

<ResponseAttach>
  <Result>This is my response attachment message</Result>
</ResponseAttach>

```

```

<doc1>
  <PrimaryDocument SCIObjectID="          :169ca65:109de7de0fd:-26e9" />
</doc1>
<attachments>
  <attachment_1>
    <document index="1">
      <PrimaryDocument SCIObjectID="          :169ca65:109de7de0fd:-26e9" />
    </document>
    <properties>
      <BodyName>BodyName1</BodyName>
      <BodyDescription>BodyDescription1</BodyDescription>
      <BodyVersion>BodyVersion1</BodyVersion>
      <BodySchema>BodySchema1</BodySchema>
      <BodyEncoding>UTF-8</BodyEncoding>
      <BodyContentType>text/xml</BodyContentType>
      <IsXMLContent>>true</IsXMLContent>
    </properties>
  </attachment_1>
</attachments>

```

Message To Service:

```

<SAPXIOut>
  <SyncWaitTimeout>60</SyncWaitTimeout>
  <BootstrapBP>sapxiout_response</BootstrapBP>
  <LocalPassword>*****</LocalPassword>
  <RemotePassword>*****</RemotePassword>
  <JNDILocalAddress>          :50004</JNDILocalAddress>

<RemoteJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl</RemoteJNDIContext>
  <ServiceType>Outbound</ServiceType>

<LocalJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl</LocalJNDIContext>
  <JNDIRemoteAddress>          :50004</JNDIRemoteAddress>
  <LocalRegistryType>EXTERNAL_REG</LocalRegistryType>
  <RemoteUser>          </RemoteUser>
  <SyncWait>Yes</SyncWait>
  <LocalUser>          </LocalUser>
  <JNDILocalKey>com.sterlingcommerce.woodstock.services.sapxigareturn</JNDILocalKey>
  <XMLVersion>1.0</XMLVersion>

<JNDIRemoteKey>com.sterlingcommerce.woodstock.services.sapxirareturn</JNDIRemoteKey>
  <XICallType>SYNCHRONOUS_RESPONSE</XICallType>
  <OriginatorID>          .169ca65.109de7de0fd.F5647</OriginatorID>
  <AttachmentRoot>attachments</AttachmentRoot>
  <FromParty/>
  <FromService>I03_OutboundToGIS</FromService>
  <ToParty/>
  <ToService>I03_InboundFromGIS</ToService>
  <Action>CarstensMessageInterfaceSync</Action>
  <ActionNS>http://stercomm.com/adapters/gis</ActionNS>
  <BodyName>BodyName</BodyName>
  <BodyDescription>BodyDescription</BodyDescription>
  <BodyVersion/>
  <BodySchema/>

```

```

<BodyEncoding>utf-8</BodyEncoding>
<BodyContentType>text/xml</BodyContentType>
<IsXMLContent>>true</IsXMLContent>
</SAPXIOut>

```

BPML Business Process (Outbound):

```

<process name="sapxiout_response">
  <sequence name="sap">
    <assign to="ResponseAttach/Result" from="'This is my response attachment
message'" />
    <operation name="XML Encoder">
      <participant name="XMLEncoder" />
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">process_data_to_document</assign>
        <assign to="root_element">ResponseAttach</assign>
        <assign to="XPath" from="'ResponseAttach/node()'" />
        <assign to="ResponseAttach" from="ResponseAttach/node()" />
      </output>
      <input message="inmsg">
        <assign to="doc1" from="*" />
      </input>
    </operation>
    <!-- Create Response Payload (overwrite Prim. Doc.) -->
    <assign to="ResponsePay/Result" from="'This is my response payload message'" />

    <operation name="XML Encoder">
      <participant name="XMLEncoder" />
      <output message="XMLEncoderTypeInputMessage">
        <assign to="mode">process_data_to_document</assign>
        <assign to="root_element">ResponsePay</assign>
        <assign to="XPath" from="'ResponsePay/node()'" />
        <assign to="ResponsePay" from="ResponsePay/node()" />
      </output>
      <input message="inmsg">
        <assign to="." from="*" />
      </input>
    </operation>
    <!-- Create Response Attachment structure -->
    <assign to="attachments/attachment_1/document" from="doc1/node()"></assign>
    <assign to="attachments/attachment_1/document/@index" from="'1'"></assign>
    <assign to="attachments/attachment_1/properties/BodyName"
from="'BodyName1'"></assign>
    <assign to="attachments/attachment_1/properties/BodyDescription"
from="'BodyDescription1'"></assign>
    <assign to="attachments/attachment_1/properties/BodyVersion"
from="'BodyVersion1'"></assign>
    <assign to="attachments/attachment_1/properties/BodySchema"
from="'BodySchema1'"></assign>
    <assign to="attachments/attachment_1/properties/BodyEncoding"
from="'UTF-8'"></assign>
    <assign to="attachments/attachment_1/properties/BodyContentType"
from="'text/xml'"></assign>
    <assign to="attachments/attachment_1/properties/IsXMLContent"
from="'true'"></assign>
    <!--Now send back response with outbound adapter -->
    <operation name="SAPXI_DeliveryResponse">

```

```

<participant name="GetFromXI" />
<output message="SAPXIOut">
<assign to="XICallType" from="'SYNCHRONOUS_RESPONSE'"></assign>
<assign to="OriginatorID" from="OriginatorID/text()"></assign>
<assign to="AttachmentRoot" from="'attachments'"></assign>
<assign to="FromParty" from="SAPXIOutbound/Header/ToParty/text()" />
<assign to="FromService" from="SAPXIOutbound/Header/ToService/text()" />
<assign to="ToParty" from="SAPXIOutbound/Header/FromParty/text()" />
<assign to="ToService" from="SAPXIOutbound/Header/FromService/text()" />
  <assign to="Action" from="'CarstensMessageInterfaceSync'"></assign>
  <assign to="ActionNS" from="'http://stercomm.com/adapters/gis'"></assign>
<assign to="BodyName"
from="SAPXIOutbound/properties/BodyName/text()"></assign>
  <assign to="BodyDescription"
from="SAPXIOutbound/properties/BodyDescription/text()"></assign>
  <assign to="BodyVersion"
from="SAPXIOutbound/properties/BodyVersion/text()"></assign>
  <assign to="BodySchema"
from="SAPXIOutbound/properties/BodySchema/text()"></assign>
  <assign to="BodyEncoding"
from="SAPXIOutbound/properties/BodyEncoding/text()"></assign>
  <assign to="BodyContentType"
from="SAPXIOutbound/properties/BodyContentType/text()"></assign>
  <assign to="IsXMLContent"
from="SAPXIOutbound/properties/IsXMLContent/text()"></assign>
  <assign to="IsXMLContent" from="'true'"></assign>
  <assign to="AttachmentRoot" from="'attachments'"></assign>
<assign to="." from="*"></assign>
</output>
<input message="SAPXIResponse">
  <assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</process>

```

Parameters Passed From Business Process to Adapter

The following table contains the parameters passed from the business process to the SAP XI adapter:

Parameter	Description
XICallType	Specifies the message handling mode. Required. Valid value: <ul style="list-style-type: none"> ◆ SYNCHRONOUS – Use for an inbound adapter ◆ SYNCHRONOUS_RESPONSE – Use to pass back a synchronous response with an outbound adapter
AttachmentRoot	If set, the parameter refers to an XML structure in the process data containing the attachment documents of the message. If omitted, the root "attachments" is used. Sending attachments in a message is optional. Optional. Valid value is a string that defines the Root Tag. Default is attachments.

Parameter	Description
ToParty	ToParty field for the SAP XI Message envelope. May be required or optional depending on the SAP XI scenario.
FromParty	FromParty field for SAP XI Message Envelope. May be required or optional depending on the SAP XI scenario. String.
ToService	ToService field for SAP XI Message Envelope. Required. String.
FromService	FromService field for SAP XI Message Envelope. Required. String.
Action	Action field for SAP XI Message Envelope. Required. String.
ActionNS	Action Name Space field for XI Message Envelope. Required. String.
MessageID	Unique Message ID on Application side for XI Message Envelope. Optional. If omitted, a GUID is generated automatically. Default is GUID.
MessageDescription	Message Description field for XI Message Envelope. Optional. String.
BodyName	Body name of the SAP XI Message. Optional. String.
BodyDescription	Body description of the SAP XI Message. Optional. String.
BodyVersion	Body version of the SAP XI Message. Optional. String.
BodySchema	XML only. Body name of the SAP XI message. String. Recommended.
BodyEncoding	Character set used for the SAP XI message. String. Default is UTF-8. Recommended.
BodyContentType	Body content type of the SAP XI message. String. Optional.
IsXMLContent	Specifies whether the SAP XI message content is in XML format. Boolean string. Recommended.
OriginatorID	Outbound message processing only. Uniquely identifies the outbound session waiting for the synchronous response. OriginatorID is generated automatically by the outbound adapter in process data and must be passed back in the response call. Required.

Script Adapter

The following table provides an overview of the Script adapter:

System name	BSF
Graphical Process Modeler (GPM) category	All Services
Description	Enables Application to execute scripts written in various languages that utilize the Bean Scripting Framework (BSF), such as Jython and JPython. For more information about the Bean Scripting Framework and supported languages, see http://jakarta.apache.org/bsf/ .
Business usage	Executes a script written in a language supported by the Bean Scripting Framework. If a customer has in-house expertise in one of the supported script languages, this adapter can be used to quickly solve the business need without having to write a new adapter.
Usage example	A customer already has scripts in place to perform some processing task and wants to incorporate that into a business process. As long as the script is written in a language that is supported by the Bean Scripting Framework, then it can be easily incorporated using this adapter.
Preconfigured?	No
Requires third party files?	At a minimum, the BSF classes (bsf.jar) are required, plus any and all script language classes (for example, jython.jar, bsh.jar). For more information about jython.jar, see www.jython.org . For more information about bsf.jar, see http://jakarta.apache.org/bsf/ .
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	Can be performed but is up to the individual script writer.
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	The script can set the returned status, therefore the individual script writer determines what status is set. The only exception is when the script runs and no status is set; in this case the adapter returns a status of Success.
Restrictions	This adapter has been tested with Python only. Scripts written in other supported languages may work, but be certain that the language comes with the necessary classes and that you have access to its documentation.

How the Script Adapter Works

The Script adapter gives users a way to incorporate scripts into Application business processes without having to write a new adapter to handle the business need.

The Script adapter executes scripts written in languages supported by the Bean Scripting Framework. The Script adapter provides tight coupling between BSF user scripts and Application by allowing scripts to access java objects that are normally only available to Application adapters, thus providing a solution to complex business problems.

Example

Your company uses a Python script to generate midday inventory reports from a legacy tracking system residing on a mainframe. Application runs under UNIX, receiving and processing incoming orders and producing midday order reports. Rather than using two systems to produce the midday status reports, you use the Script adapter to import inventory data into the order reporting business process, allowing for easier tracking of orders against existing inventory.

Business Process Example

The following example uses the Script adapter:

```
<process name = "BSF">
  <operation name="Script Adapter">
    <participant name="ScriptAdapter"/>
    <output message="ScriptAdapterInputMessage">
      <assign to="scriptFileName"/>somepath/test.py</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

Script Adapter Restrictions

The only scripts this adapter can execute have to be supported by the Bean Scripting Framework. If and when new script languages develop script engines that work with the Bean Scripting Framework, no code changes need to occur in the adapter code, as new script languages can be registered and used at any time.

Note: Most of this technology is open source, which brings with it certain caveats. Refer to the appropriate open source site for support.

Third-Party Adapters

For some adapters used with third-party software, you must install specific .jar files or drivers before you can use the adapter with Application. These third-party .jar files are available from the applicable third-party software vendors. Often, you can download the files from the vendor's Web site. If an adapter requires any third-party files, the specific information about the files is included in the documentation for that adapter.

To install a third party .jar file:

1. Copy the .jar file to a directory on the host computer where Application is installed. Record the path and name of the directory.
2. Shut down Application if it is running.
3. Install the .jar file by running the install3rdParty script located in the *installdir/bin* directory. You can access the following instructions for using the command by typing in “install3rdParty” at the command line:

```
Usage: ./install3rdParty.sh vendorName vendorVersion <-j | -l | -p | -r |
-d > filelist [-nodeploy]
```

using:

```
-j for jar/zip files
-l for shared libraries
-p for properties files
-r for resource properties files
-d for database jar/zip files
```

```
./install3rdParty.sh ibm_sap 3_5 -j /usr/local/lib/sap/*.jar
./install3rdParty.sh ibm_sap 3_5 -j /usr/local/lib/sap/infobus.jar
./install3rdParty.sh oracle 1_2 -d /usr/local/lib/jar/*.jar
./install3rdParty.sh oracle 1_2 -d /usr/local/lib/jar/classes12.zip
./install3rdParty.sh ibm_sap 3_5 -j /usr/local/lib/sap/infobus.jar
./install3rdParty.sh ibm_sap 3_5 -l /usr/local/lib/sap/*.so
./install3rdParty.sh ibm_sap 3_5 -l
/usr/local/lib/sap/libivjsid35.so
./install3rdParty.sh ibm_sap 3_5 -p /usr/local/lib/sap/*.properties
./install3rdParty.sh ibm_sap 3_5 -p
/usr/local/lib/sap/some.properties
./install3rdParty.sh mqseris 2_0 -r
/usr/local/lib/mqs/service.properties
```

4. Restart Application. You can now use this adapter in Application.

Implementing the Script Adapter

To implement the Script adapter, complete the following tasks:

1. Write a script in a language supported by the Bean Scripting Framework.
2. Configure the predefined Script adapter, or create a new Script adapter configuration. For information, see *Managing Services and Adapters*.
3. Configure the Script adapter. For information, see *Configuring the Script Adapter* on page 1370.
4. Test the Script adapter configuration. For information, see *Testing the Script Adapter Configuration* on page 1370.

5. Use the Script adapter in a business process.

Configuring the Script Adapter

To configure the Script adapter, you must specify settings for the following fields in the GPM:

Field	Description
Script Filename (scriptFileName)	Fully qualified path and filename of the script to execute. Required.
Script Language Name (langName)	Name of the script language (for example, Python). Optional — only required if not already included within Bean Scripting Framework (BSF).
Script Engine Class (engineClass)	The name of the script engine class (for example, org.apache.bsf.engines.jpython.JPythonEngine). Optional — only required if not already included within BSF.
Script Language Extension (langExt)	The extension used to associate with this language (for example: py, pl, or bsh). Optional — only required if not already included within BSF.

Objects Exposed for Use By a Script (Object Name – Class Exposed)

The following objects are available for a script to use:

- siWfc – WorkflowContext
- siIwf – InitialWorkflowContext
- siWfd – WorkflowDef
- siRmi – BSFServer (the “little-a” RMI class for doing I/O)
- siDoc – Document
- siLog - XLogger

Testing the Script Adapter Configuration

After you have a script written that you want to test, create an instance of the Script adapter and create a business process that runs that instance.

Python Script Example

The following example (Test.py) illustrates a script written in Python:

```
# This script expects an input file with the workflow context
# This script expects a bp called zze (FS Extract) to exist
siDoc = siWfc.getPrimaryDocument();
name = siDoc.getBodyName();
body = siDoc.getBody();
siLog.logError(name);
stuff = siRmi.readFile("/home/test.txt");
idl = siWfd.getIDForName("zze");
siIwf.setWorkflowDefId(idl);
```

```
siIwf.setDocumentName("junk.txt");
siIwf.setDocumentBody(body);
siWfk = siIwf.start();
id2 = siWfk.getWorkflowId();
siLog.logError(str(id2));
siWfc.addBootStrapWorkFlows(str(id2));
siDoc = siWfc.createDocument();
siDoc.setBody("this is a test");
siDoc.setBodyName("test.txt");
siWfc.putPrimaryDocument(siDoc);
siWfc.setWFContent("TestVal", "Hello");
siWfc.setWFStatusRpt("Status_Report", "hello");
siWfc.setAdvancedStatus("worked");
siWfc.setBasicStatus(siWfc.SUCCESS);
```

Self Registration Service

The following table provides a high-level overview of the Self Registration Service:

System name	Self Registration Service
Graphical Process Modeler (GPM) categories	SyncMode > SelfRegistrationType3_1
Description	This service is used to add a user, associate a user with a group, and update user information or change a user's password.
Business usage	The Self Registration Service allows you to add a user, associate a user with a group, and update user information or change a user's password in the system.
Usage example	A Gentran Integration Suite business process is executed that requires you to either add a user, associate a user with a group, update user information or change a user's password in database. Gentran Integration Suite uses the Self Registration Service to add a user, associate a user with a group, update user information or change a user's password in the system.
Preconfigured?	The Self Registration Service must be installed and deployed before it can be invoked.
Requires third party files?	No
Platform availability	This adapter is available on these platforms: <ul style="list-style-type: none">◆ Microsoft Windows 2000◆ Sun Solaris◆ HP-UX◆ IBM-AIX◆ United Linux◆ RedHat AS◆ iSeries (OS/400)◆ zSeries (z/OS)
Related services	Pre 3.1 Self Registration Service
Application requirements	The user must exist in the system for associating a user with a group, updating user information, or changing a user's password.
Initiates business processes?	None
Invocation	Runs as part of a business process.

Business process context considerations	The user must exist in the system for associating a user with a group, updating user information, or changing a user's password.
Returned status values	<ul style="list-style-type: none"> ◆ 0 - Success ◆ 1 - Error
Restrictions	None
Persistence level	Default
Testing considerations	Debug information for this service can be found in the system log files.

Configuration Parameters

Parameter (Display Name)	Description
Name	The name that the service will have in Gentran Integration Suite.
Description	Description of the service.
Select a Group	<p>A button that allows selection or creation of the group that this service will be part of.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ None ◆ Create New Group ◆ Select Group

Parameters Passed from Business Process to Service

Parameter (Display Name)	Description
mode	The mode used to run the Self Registration Service for a user.
username	<p>The username of the password you want to update.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Any valid existing user name
password	<p>The password for the new user.</p> <ul style="list-style-type: none"> ◆ Any valid password
oldPassword	<p>The old username password.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Only valid current password

Parameter (Display Name)	Description
newPassword	The new password you want to update for the username. Valid values: ◆ Any valid new password
retypedPassword	The new password you want to update for the username. Valid values: ◆ Any valid new password
fname	The first name for the new/existing user. Valid values: ◆ Any valid user first name
lname	The last name for the new/existing user. Valid values: ◆ Any valid user last name
timeout	The timeout for the new/existing user. Valid values: ◆ Any valid integer greater than 0
group	The group name for adding the user to this group. Valid values: ◆ Any valid existing group
status	The activate status for updating the existing user. Valid values: ◆ Any valid user activate status, (1=ENABLED; 0=DISABLED)
email	The email for updating the existing user. Valid values: ◆ Any valid user email
pager	The pager for updating the existing user. Valid values: ◆ Any valid new password.

Example Business Process

addUser Business Process Example

```

<!-- For addUser mode -->
<process name="than">
  <sequence name="testuser">
    <operation name="testuser">
      <participant name="SelfRegistration3_1"/>
      <output message="Xout">

```

```

        <assign to="mode">addUser</assign>
        <assign to="username">testuser</assign>
        <assign to="password">P@$$Word</assign>
        <assign to="fname">firstname</assign>
        <assign to="lname">lastname</assign>
        <assign to="timeout">100</assign>
    </output>
    <input message="Xin">
</input>
</operation>
</sequence>
</process>

```

addUserToGroup Business Process Example

```

<!-- For addUserToGroup mode -->
<process name="testuser">
    <sequence name="testuser">
        <operation name="testuser">
            <participant name="SelfRegistration3_1"/>
            <output message="Xout">
                <assign to="mode">addUserToGroup</assign>
                <assign to="username">testuser</assign>
                <assign to="group">testgroup</assign>
            </output>
            <input message="Xin">
</input>
        </operation>
    </sequence>
</process>

```

updateUserInfo Business Process Example

```

<!-- For updateUserInfo mode -->
<process name="testuser">
    <sequence name="testuser">
        <operation name="testuser">
            <participant name="SelfRegistration3_1"/>
            <output message="Xout">
                <assign to="mode">updateUserInfo</assign>
                <assign to="username">testuser</assign>
                <assign to="status">0</assign>
                <assign to="fname">firstname</assign>
                <assign to="lname">lastname</assign>
                <assign to="email">test@test.com</assign>
                <assign to="pager">123-4567</assign>
                <assign to="timeout">100</assign>
            </output>
            <input message="Xin">
</input>
        </operation>
    </sequence>
</process>

```

changePassword Business Process Example

```

<!-- For changePassword mode -->
<process name="testuser">

```

```
<sequence name="testuser">
  <operation name="testuser">
    <participant name="SelfRegistration3_1"/>
    <output message="Xout">
      <assign to="mode">changePassword</assign>
      <assign to="username">testuser</assign>
      <assign to="oldPassword">P@$$Word</assign>
      <assign to="newPassword">NewP@$$Word</assign>
      <assign to="retypedPassword">NewP@$$Word</assign>
    </output>
    <input message="Xin">
      </input>
    </operation>
  </sequence>
</process>
```

SFTP Client Adapter

The SFTP Client adapter has the following major features:

Uses perimeter services

Commands are scriptable through BPML

Works easily with most SFTP servers

Accessible through the following services in a business process:

- ◆ SFTP Client Begin Session service
- ◆ SFTP Client CD service
- ◆ SFTP Client DELETE service
- ◆ SFTP Client End Session service
- ◆ SFTP Client GET service
- ◆ SFTP Client LIST service
- ◆ SFTP Client MOVE service
- ◆ SFTP Client PUT service
- ◆ SFTP Client PWD service

The following table provides an overview of the SFTP Client adapter:

System name	SFTP Client Adapter
Graphical Process Modeler (GPM) category	None
Description	Sends SFTP requests to trading partners through perimeter services.
Business usage	Use this adapter to send SFTP requests to perform activities such as to <i>get</i> or <i>put</i> files into a directory on the trading partner's SFTP server.
Usage example	A business process is executed that translates a document to send to a trading partner. After the translation, the SFTP Client adapter sends the document to the trading partner.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All Application supported platforms

Related services	<p>The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations:</p> <ul style="list-style-type: none"> ◆ SFTP Client Begin Session service ◆ SFTP Client CD service ◆ SFTP Client DELETE service ◆ SFTP Client End Session service ◆ SFTP Client GET service ◆ SFTP Client LIST service ◆ SFTP Client MOVE service ◆ SFTP Client PUT service ◆ SFTP Client PWD service
Application requirements	<p>An SFTP Server at the external trading partner location.</p> <p>When this adapter is configured with a 'non-local-mode' perimeter server, the perimeter server must be installed and running. The perimeter server is typically installed in a DMZ environment, separated from Application by a firewall. Refer to the perimeter services documentation for details on installing and running that component.</p>
Initiates business processes?	No
Invocation	This adapter is not invoked from a business process.
Business process context considerations	Business processes using SFTP client services should not be marked as Auto Resume. These services require an established session which will no longer exist after a restart.
Returned status values	None
Restrictions	None
Persistence level	N/A
Testing considerations	<p>To test this adapter, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see the <i>Business Process Example of All SFTP Client Services</i> on page 1380.</p> <p>Debug information for this adapter is located at:</p> <p>Operations > System > Logs > SFTP Client Adapter and Services</p>

Implementing the SFTP Client Adapter

To implement the SFTP Client adapter, complete the following tasks:

1. Create an SFTP Client adapter configuration. See *Managing Services and Adapters*.
2. Configure the SFTP Client adapter. See *Configuring the SFTP Client Adapter* on page 1379.

Configuring the SFTP Client Adapter

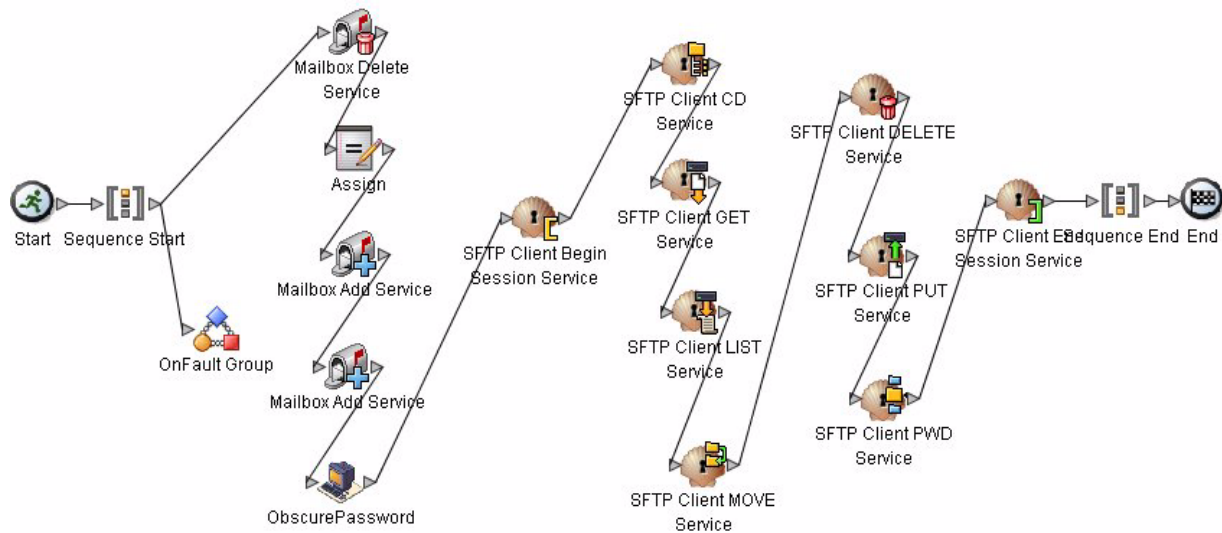
To configure the SFTP Client adapter:

1. Select **Deployment > Services > Configuration**.
2. Search for SFTP Client adapter or select it from the list.
3. Click **Edit**.
4. Specify field settings:

Field	Description
Name	Name this adapter will have in Application
Description	Description of adapter
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .
Perimeter Server	List of perimeter servers, including local-mode perimeter servers. Required.
Minimum Number of Threads	A tuning parameter that indicates the range of threads available for handling events to improve performance. See Performance and Tuning for more information. Must be less than or equal to Maximum Number of Threads value. Required.
Maximum Number of Threads	A tuning parameter that indicates the range of threads available for handling events to improve performance. See Performance and Tuning for more information. Must be greater than or equal to Minimum Number of Threads value. Required.
LocalPortRange	Any valid port number(s) not being used by another application running on the system. A port in the specified range will be used to establish an SSH channel to the remote SFTP server. Optional. Valid values are: (empty) - system selects an available port 0 - system selects an available port <i>nnnn</i> - use specified port <i>nnnn</i> , for example 9012 <i>nn-yy</i> - use a port in the range of <i>nn</i> to <i>yy</i> , for example 462-863 would use a port in the range of 462 to 863, inclusive Multiples of the above values can be specified, separated by commas, for example 9012, 462-863, 4925

Business Process Example of All SFTP Client Services

To help you get started using the SFTP Client adapter and SFTP Server adapter, Application includes a demo that provides an example of all the services. The demo transfers a file from the SFTP Client adapter to the SFTP Server adapter. The following graphic is the business process model of SFTPClientDemoAllServices:



The BPML code associated with SFTPClientDemoAllServices is shown below:

```
<!-- Import service initializes mailboxes. Creating mailboxes, permissions and
Virtual Root if they do not exist.-->
<operation name="Import Service">
  <participant name="ImportService"/>
  <output message="ImportTypeInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="ImportServiceResults" from="*"></assign>
  </input>
</operation>

<!-- Mailbox Delete Service deletes messages in all mailboxes below virtual root
mailbox. MailboxPath is commented out but could be used to delete messages in a
specific mailbox. -->

<operation name="Mailbox Delete Service">
  <participant name="MailboxDelete"/>
  <output message="MailboxDeleteServiceTypeInputMessage">
    <assign to="MessageNamePattern">message*</assign>
    <assign to="UserId">admin</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="MailboxDeleteServiceResults" from="*"></assign>
  </input>
</operation>
```

```

</operation>

<!-- Mailbox Add Service adds Primary Document as message1 in mailbox sub1 -->
<operation name="Mailbox Add Service">
  <participant name="MailboxAdd"/>
  <output message="MailboxAddServiceTypeInputMessage">
    <assign to="MailboxPath"/>/sub1</assign>
    <assign to="MessageName">message1</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="MailboxAddServiceResults" from="*"></assign>
  </input>
</operation>

<!-- Mailbox Add Service adds Primary Document as message2 in mailbox sub1 -->
<operation name="Mailbox Add Service">
  <participant name="MailboxAdd"/>
  <output message="MailboxAddServiceTypeInputMessage">
    <assign to="MailboxPath"/>/sub1</assign>
    <assign to="MessageName">message2</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="MailboxAddServiceResults" from="*"></assign>
  </input>
</operation>

<!-- Get obscured password -->
<operation name="Obscure Password">
  <participant name="FTPClientObscureParameter"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<!-- Begin Session -->
<operation name="SFTP Client Begin Session Service">
  <participant name="SFTPClientBeginSession"/>
  <output message="SFTPClientBeginSessionServiceTypeInputMessage">
    <assign to="SFTPClientAdapter">SFTPClientAdapter</assign>
    <assign to="RemoteHost">carvey</assign>
    <assign to="RemotePasswd" from="admin/text()"></assign>
    <assign to="RemotePort">60849</assign>
    <assign to="RemoteUserId">ftpdemo</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="SFTPClientBeginSessionServiceResults" from="*"></assign>
  </input>
</operation>

<!-- SFTP Client PWD Service returns current directory -->

```

```

    <operation name="SFTP Client PWD Service">
      <participant name="SFTPClientPwd"/>
      <output message="SFTPClientPwdServiceTypeInputMessage">
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="SFTPClientPwdServiceResults" from="*"></assign>
      </input>
    </operation>

<!-- SFTP Client LIST Service returns directories and files under tag ListNames -->
    <operation name="SFTP Client LIST Service">
      <participant name="SFTPClientList"/>
      <output message="SFTPClientListServiceTypeInputMessage">
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="SFTPClientListServiceResults" from="*"></assign>
      </input>
    </operation>

<!-- SFTP Client SFTP CD SERVICE changes to directory assigned to tag Directory -->
    <operation name="SFTP Client SFTP CD SERVICE">
      <participant name="SFTPClientCd"/>
      <output message="CdRequest1">
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <!--<assign to="Directory"
from="SFTPClientListServiceResults/ListNames/Name[1]/text()"></assign>-->
        <assign to="Directory"
from="SFTPClientListServiceResults/Files/File[1]/Name/text()"></assign>
      </output>
      <input message="inmsg">
        <assign to="SFTPClientCdServiceResults" from="*"></assign>
      </input>
    </operation>

<!-- SFTP Client LIST Service returns directories and files under tag ListNames -->
    <operation name="SFTP Client LIST Service">
      <participant name="SFTPClientList"/>
      <output message="SFTPClientListServiceTypeInputMessage">
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="SFTPClientListServiceResults2" from="*"></assign>
      </input>
    </operation>

```

```

<!-- SFTP Client GET Service retrieves files specified by RemoteFilePattern or
RemoteFileName from current directory or from value identified by tag RemoteDirectory
if specified. -->
  <operation name="SFTP Client GET Service">
    <participant name="SFTPClientGet"/>
    <output message="SFTPClientGetServiceTypeInputMessage">
      <assign to="RemoteFileName"

from="SFTPClientListServiceResults2/Files/File[Name/text()='message2']/Name/text()">
</assign>
      <assign to="SessionToken"

from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    </output>
    <input message="inmsg">
      <assign to="SFTPClientGetServiceResults" from="*"></assign>
    </input>
  </operation>

<!-- SFTP Client SFTP CD SERVICE changes to directory assigned to tag Directory -->
  <operation name="SFTP Client SFTP CD SERVICE">
    <participant name="SFTPClientCd"/>
    <output message="CdRequest1">
      <assign to="Directory">../sub2</assign>
      <assign to="SessionToken"

from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    </output>
    <input message="inmsg">
      <assign to="SFTPClientCdServiceResults2" from="*"></assign>
    </input>
  </operation>

<!-- SFTP Client PUT Service performs the ftp command put. The Primary Document
contains the files to be put. RemoteFileDirectory is omitted the files are placed in
the current remote directory. -->
  <operation name="SFTP Client PUT Service">
    <participant name="SFTPClientPut"/>
    <output message="SFTPClientPutServiceTypeInputMessage">
      <assign to="SessionToken"

from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
      <assign to="." from="SFTPClientGetServiceResults/DocumentId" append="true"/>
    </output>
    <input message="inmsg">
      <assign to="SFTPClientPutServiceResults" from="*"></assign>
    </input>
  </operation>

<!-- SFTP Client Move Service performs the ftp move command renaming message1 to
message1_renamed in the same directory. -->
  <operation name="SFTP Client MOVE Service">
    <participant name="SFTPClientMove"/>
    <output message="SFTPClientMoveServiceTypeInputMessage">
      <assign to="RemoteFromFileName">message2</assign>
      <assign to="RemoteToFileName">message2_renamed</assign>
      <assign to="SessionToken"

from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    </output>
  </operation>

```

```

        <assign to="." from="SFTPClientGetServiceResults/DocumentList"
append="true"/>
    </output>
    <input message="inmsg">
        <assign to="SFTPClientMoveServiceResults" from="*"></assign>
    </input>
</operation>

<!-- SFTP Client SFTP CD SERVICE changes to directory assigned to tag Directory -->
<operation name="SFTP Client SFTP CD SERVICE">
    <participant name="SFTPClientCd"/>
    <output message="CdRequest1">
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="Directory">..</assign>
    </output>
    <input message="inmsg">
        <assign to="SFTPClientCdServiceResults3" from="*"></assign>
    </input>
</operation>

<!-- SFTP Client Move Service performs the ftp move command moving message from
mailbox sub1 to mailbox sub2. -->
<operation name="SFTP Client MOVE Service">
    <participant name="SFTPClientMove"/>
    <output message="SFTPClientMoveServiceTypeInputMessage">
        <assign to="RemoteFromFileName">sub2/message2_renamed</assign>
        <assign to="RemoteToFileName">sub3/message2</assign>
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    </output>
    <input message="inmsg">
        <assign to="SFTPClientMoveServiceResults2" from="*"></assign>
    </input>
</operation>

<!-- SFTP Client Delete Service removes files from directories. In this case removes
message1 from sub1 -->
<operation name="SFTP Client DELETE Service">
    <participant name="SFTPClientDelete"/>
    <output message="DeleteRequest">
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="RemoteFileName">sub3/message2</assign>
    </output>
    <input message="inmsg">
        <assign to="SFTPClientDeleteServiceResults" from="*"></assign>
    </input>
</operation>

<!-- SFTP Client LIST Service returns directories and files under tag ListNames -->
<operation name="SFTP Client LIST Service">
    <participant name="SFTPClientList"/>
    <output message="SFTPClientListServiceTypeInputMessage">
<assign to="RemoteFileName">sub2/*</assign>

```



```

        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
        <assign to="SFTPClientListServiceResults3" from="*"></assign>
    </input>
</operation>

<!-- SFTP Client End Session Service, ends session specified by SessionToken.-->
<operation name="SFTP Client End Session Service">
    <participant name="SFTPClientEndSession"/>
    <output message="SFTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
        <assign to="SFTPClientEndSessionServiceResults" from="*"></assign>
    </input>
</operation>

<!-- To catch all onFault handling to log an error to PsSftpClient -->
<onFault>
    <sequence name="End Session">
        <operation name="SFTP Client End Session Service">
            <participant name="SFTPClientEndSession"/>
            <output message="SFTPClientEndSessionServiceTypeInputMessage">
                <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
                <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
                <assign to="SFTPClientEndSessionServiceOnFaultResults"
from="*"></assign>
            </input>
        </operation>
    </sequence>
</onFault>

```

Activity Monitoring for the SFTP Client Adapter

The SFTP Client adapter creates activity monitoring records for the following activities:

- Active sessions

- In progress PUTs display the data transferred in kbps with a progress indicator

- In progress GETs display the data transferred in kbps

Select **Business Processes > Current Activities > SFTP Client Adapter**.

SFTP Client Begin Session Service

The following table provides an overview of the SFTP Client Begin Session service:

System name	SFTP Client Begin Session Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Starts an SFTP session with an external trading partner for the purpose of exchanging business documents. This service works through an instance of the SFTP Client adapter.
Business usage	Use this service to establish a session with a trading partner SFTP server.
Usage example	A business process executes to translate a document sent to a trading partner. The trading partner profile specifies SFTP as the transport protocol. Application uses the SFTP Client Begin Session service to establish a session with the trading partner's SFTP server.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	<p>The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations:</p> <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service <p>To mask the values associated with the remote password parameter, use the Obscure Data - Process Data Values service in conjunction with the SFTP Client Begin Session service. This service is presented in GPM as Obscure Parameter on the All Services stencil. Then, the password must be revealed with the revealObscured XPath function before sending it to the service.</p>
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.

Business process context considerations	The SFTP Client Begin Session service allows you to specify a remote password. To obscure this password in process data for the business process, use the Obscure Data - Process Data Values service within the same business process (GPM display name is Obscure Parameter). The Obscure Data - Process Data Values service masks the values associated with parameters.
Returned status values	0–Success 1– Error
Restrictions	N/A
Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see <i>SFTP Client adapter</i> . To view debug information for this service: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The values for many of the parameters for the Begin Session service can be specified from multiple sources. The source for each parameter is determined by answering the following questions:

Is the parameter specified in the business process?

Yes - Application uses those values.

No - Is ProfileId specified?

Yes - Application uses values from the profile associated with that ProfileId.

No - Parameters noted in the following table take their value from the configuration of the selected SFTP Client adapter.

The following table contains the parameters passed from the business process to the SFTP Client Begin Session service:

Field	Description
Name	Name this service will have in Application
Description	Description of service
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>

Field	Description
PreferredAuthenticationMethod	<p>Method used to authenticate users. Required. Valid values are:</p> <p>password</p> <p>public key</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
CompressionMethod	<p>Specifies whether data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Required. Valid values: None, ZLIB. Default is None.</p> <p>Note: The value entered for this parameter overrides the Compression setting in the SSH Remote Profile configuration.</p>
ConnectionRetries	<p>The number of times the service will try to connect to the Trading Partner System. Connection retries occur only with TCP/IP related issues. Optional. Valid value is any numeric value. Default is 1.</p> <p>Note: The value entered for this parameter overrides the Connection Retry Count setting in the SSH Remote Profile configuration.</p> <p>While using the ConnectionRetries parameter, set the ResponseTimeout value to wait longer than the total time for RetryDelay and ConnectionRetires parameters. This setting allows the business process to remain active to perform the retries before the session times out and terminates. The following example illustrates the setting where the value of the ResponseTimeout (300) is greater than the total time taken by RetryDelay and ConnectionRetires parameters (30*5=150):</p> <pre><assign to="ResponseTimeout">300</assign> <assign to="RetryDelay">30</assign> <assign to="ConnectionRetries">5</assign></pre>
KnownHostKeyId	<p>Public key used to authenticate remote SFTP servers to the Application SFTP Client. Internal value is GUID. Required.</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
LocalPortRange	<p>Any valid port number(s) not being used by another application running on the system. A port in the specified range will be used to establish an SSH channel to the remote SFTP server. Optional. Valid values are:</p> <p>(empty) - system selects an available port</p> <p>0 - system selects an available port</p> <p><i>nnnn</i> - use specified port <i>nnnn</i>, for example 9012</p> <p><i>nn-yy</i> - use a port in the range of <i>nn</i> to <i>yy</i>, for example 462-863 would use a port in the range of 462 to 863, inclusive</p> <p>Multiples of the above values can be specified, separated by commas, for example 9012, 462-863, 4925</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>

Field	Description
PreferredCipher	<p>The cipher the client prefers to use for both client to server and server to client stream encryption. Required. Default is blowfish-cbc. Valid values are:</p> <ul style="list-style-type: none"> ◆ 3des-cbc ◆ blowfish-cbc ◆ aes256-cbc ◆ aes192-cbc ◆ aes128-cbc ◆ cast128-cbc ◆ twofish256-cbc ◆ twofish192-cbc ◆ twofish128-cbc <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
PreferredMAC	<p>The MAC the client prefers to use for stream encryption. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ hmac-sha1 ◆ hmac-md5 <p>Default is hmac-sha1.</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
ProfileId	SSH Remote profile identification. Optional. Valid value is any valid profile ID.
RemoteHost	<p>External Trading Partner host system (SFTP server IP Address or DNS name. Required if ProfileId is not specified. Use any valid IP Address or DNS name.</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
RemotePasswd	<p>SFTP remote login password. Either UserIdentityKeyId or RemotePasswd is required if ProfileId is not specified.</p> <p>Note: You can obscure the password using the Obscure service. To reveal the password for the service, use the XPath function 'revealObscured(element), where (element) contains a text value of the obscured password, obtained from the Obscure parameter service.</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
RemotePort	<p>External Trading Partner port number. Required if ProfileId is not specified.</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
RemoteUserId	<p>SFTP remote login username. Required if ProfileId is not specified.</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>

Field	Description
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.
RetryDelay	Number of seconds the adapter will wait before retrying. Optional. Valid value is any numeric value. Note: The value entered for this parameter overrides the Retry Delay setting in the SSH Remote Profile.
SFTPClientAdapter	Select the SFTP Client adapter or a group that includes an SFTP Client adapter for this service to use when beginning sessions with an SFTP server. Required.
UserIdentityKeyId	Key pair used to authenticate the remote user to the server. Either UserIdentityKeyId or RemotePasswd is required if ProfileId is not specified. Note: The value entered for this parameter overrides the User Identity Key setting in the SSH Remote Profile configuration.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client Begin Session service to the business process:

Parameter	Description
SessionToken	Specifies the identifier for the session established between the SFTP Client adapter and an SFTP server.
SessionStartTime	The date/time stamp for when the session started.
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: OK - 0 General Failure - 4 Bad Message - 5 No Connection - 6 Connection Lost - 7 Operation Unsupported - 8
ServiceStartTime	The date/time stamp for when the service started.
ServiceEndTime	The date/time stamp for when the service ended.

Business Process Example

The following example business process excerpt illustrates using the SFTP Client Begin Session service:

```
<operation name="Obscure Password">
```

```

    <participant name="FTPClientObscureParameter"/>
    <output message="outmsg">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

<operation name="SFTP Client Begin Session Service">
  <participant name="SFTPClientBeginSession"/>
  <output message="SFTPClientBeginSessionServiceTypeInputMessage">
    <assign to="SFTPClientAdapter">SFTPClientAdapter</assign>
    <assign to="RemoteHost">myhost</assign>
    <!-- Using the revealObscured XPath function ◊
    <assign to="RemotePasswd" from="revealObscured(admin)"></assign>
    <assign to="PreferredAuthenticationMethod">password</assign>
    <assign to="RemotePort">21</assign>
    <assign to="RemoteUserId">myname</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="SFTPClientBeginSessionServiceResults" from="*"></assign>
  </input>
</operation>

[[Insert SFTP Client End Session]]

```

SFTP Client CD Service

The following table provides an overview of the SFTP Client CD service:

System name	SFTP Client CD Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Changes directories on the trading partner's SFTP server.
Business usage	Use this service to change directories on the trading partner SFTP server system.
Usage example	A Application business process is executed that requires retrieving a document from the trading partner's system. The document is located in a directory other than the home directory. The SFTP Client CD service, working through the SFTP Client adapter, changes to the directory where the document is located. Then the SFTP Client GET service retrieves the document.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0–Success 1– Error
Restrictions	N/A

Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see <i>SFTP Client adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client CD service:

Field	Description
CdUp	Do not use. Note: The CdUp parameter has no affect on this service and is unsupported. To change directories to a higher level directory, use the Directory parameter for this service, and provide a value of ".."
Directory	The directory to change to. Can also be a relative directory depending on server capabilities. Required.
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is ResponseTimeout specified in SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client CD service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none">◆ OK - 0◆ No Such File - 2◆ Permission Denied - 3◆ General Failure - 4◆ Bad Message - 5◆ No Connection - 6◆ Connection Lost - 7◆ Operation Unsupported - 8
ServiceStartTime	The date/time stamp for when the service started.
ServiceEndTime	The date/time stamp for when the service ended.
PwdDirectory	Specifies the current directory.

Business Process Example

The following example business process excerpt illustrates using the SFTP Client CD service:

```
[[Insert SFTP Client Begin Session]]

<operation name="SFTP Client SFTP CD SERVICE">
  <participant name="SFTPClientCd"/>
  <output message="CdRequest1">
    <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="Directory">My Directory</assign>
  </output>
  <input message="inmsg">
    <assign to="SFTPClientCdServiceResults" from="*"></assign>
  </input>
</operation>

[[Insert SFTP Client End Session]]
```

SFTP Client DELETE Service

The following table provides an overview of the SFTP Client DELETE service:

System name	SFTP Client DELETE Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Deletes a document in a specified directory on the trading partner's SFTP server.
Business usage	Use this service to delete a document on the trading partner system when the SFTP protocol is required as the communication mechanism with the trading partner.
Usage example	A business process is executed that requires the removal of a document on the trading partner's system. The SFTP Client DELETE service, working through the SFTP Client adapter, removes the specified document from the directory on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None.
Returned status values	0–Success 1– Error
Restrictions	N/A

Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see <i>SFTP Client adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client DELETE service:

Field	Description
RemoteFileName	Name of the file to delete from the remote trading partner directory. Required.
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is ResponseTimeout specified in SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client DELETE service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none"> ◆ OK - 0 ◆ No Such File - 2 ◆ Permission Denied - 3 ◆ General Failure - 4 ◆ Bad Message - 5 ◆ No Connection - 6 ◆ Connection Lost - 7 ◆ Operation Unsupported - 8
ServiceStartTime	The date/time stamp for when the service started.
ServiceEndTime	The date/time stamp for when the service ended.

Business Process Example

The following example business process excerpt illustrates using the SFTP Client DELETE service:

```
[[Insert Begin Session]]
```

```
<operation name="SFTP Client DELETE Service">
  <participant name="SFTPClientDelete"/>
  <output message="DeleteRequest">
    <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="RemoteFileName">FilenameToDelete</assign>
  </output>
  <input message="inmsg">
    <assign to="SFTPClientDeleteServiceResults" from="*"></assign>
  </input>
</operation>
```

```
[[Insert End Session]]
```

SFTP Client End Session Service

The following table provides an overview of the SFTP Client End Session service:

System name	SFTP Client End Session Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Ends an SFTP session with an external trading partner SFTP server. This service works through an instance of the SFTP Client adapter.
Business usage	Use this service as the last functional activity in a business process that sends an SFTP request to a trading partner. This service can only be used if the SFTP Client Begin Session service is used previously in the business process.
Usage example	A business process is executed to translate a document to send to a trading partner. After the translation, Application begins a session with the trading partner using the SFTP Client adapter, sends the document, then ends the session using the SFTP Client End Session service.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process. Note: Ensure business processes using the SFTP Client Begin Session service always call SFTP Client End Session service, even in error situations. If the End Session service is not called, the session will remain visible in the Service Activity Monitor until Application is restarted.

Business process context considerations	None
Returned status values	0–Success 1– Error
Restrictions	N/A
Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see <i>SFTP Client adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client End Session service:

Field	Description
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is ResponseTimeout specified in SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service that initiated the session to be ended. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client End Session service to the business process:

Parameter	Description
SessionEndTime	The date/time stamp for when the session ended.

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none"> ◆ OK - 0 ◆ General Failure - 4 ◆ Bad Message - 5 ◆ No Connection - 6 ◆ Connection Lost - 7 ◆ Operation Unsupported - 8
ServiceStartTime	The date/time stamp for when the service started.
ServiceEndTime	The date/time stamp for when the service ended.
SessionStartTime	The date/time stamp for when the session started.

Business Process Example

The following example business process excerpt illustrates using the SFTP Client End Session service:

[[Insert Begin Session]]

```
<operation name="SFTP Client End Session Service">
  <participant name="SFTPClientEndSession"/>
  <output message="SFTPClientEndSessionServiceTypeInputMessage">
    <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="SFTPClientEndSessionServiceResults" from="*"></assign>
  </input>
</operation>
```

SFTP Client GET Service

The following table provides an overview of the SFTP Client GET service:

System name	SFTP Client GET Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Retrieves a document from a specified directory on the trading partner's SFTP server.
Business usage	Use this service to retrieve a document from a trading partner and move it into Application when the SFTP protocol is required as the transport mechanism.
Usage example	A business process is executed to retrieve a specified file from the external trading partner. Application uses the SFTP Client GET service, working through the SFTP Client adapter, to retrieve the file from a specified directory on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0–Success 1– Error

Restrictions	N/A
Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see <i>SFTP Client adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client GET service:

Field	Description
RemoteFileName	The name of the file to be retrieved from the remote trading partner. Required.
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is the ResponseTimeout value specified in the SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client GET service to the business process:

Parameter	Description
ServiceStartTime	The date/time stamp for when the service started.
DocumentId	Provides information about the file retrieved as a result of the GET service.

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none"> ◆ OK - 0 ◆ End of File - 1 ◆ No Such File - 2 ◆ Permission Denied - 3 ◆ General Failure - 4 ◆ Bad Message - 5 ◆ No Connection - 6 ◆ Connection Lost - 7 ◆ Operation Unsupported - 8
ServiceEndTime	The date/time stamp for when the service ended.
Primary Document	The file retrieved as a result of the GET service.

Business Process Example

The following example business process excerpt illustrates using the SFTP Client GET service:

[[Insert begin session here]]

```
<operation name="SFTP Client GET Service">
  <participant name="SFTPClientGet"/>
  <output message="SFTPClientGetServiceTypeInputMessage">
    <assign to="RemoteFileName" >FilenameToGet</assign>
    <assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
  </output>
  <input message="inmsg">
    <assign to="SFTPClientGetServiceResults" from="*"></assign>
  </input>
</operation>
```

[[Insert end session here]]

Correlations and Document Tracking

The following table details the correlations available from the SFTP Client GET service for document tracking:

Key	Values
ACTION	Get, Put
Direction	Inbound, Outbound

Key	Values
Protocol	SFTP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost
Username	username
RemoteFile	filename

SFTP Client LIST Service (Build 4300 - Build 4324)

The following table provides an overview of the SFTP Client LIST service:

System name	SFTP Client LIST Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Retrieves a list of files on a specified directory on the trading partner's SFTP server.
Business usage	Use this service to retrieve a list of files in a specified directory on the trading partner's system and return the list to Application when the SFTP protocol is required as the transport mechanism.
Usage example	A business process is executed to retrieve a list of files from the external trading partner. Application uses the SFTP Client LIST service, working through the SFTP Client adapter, to retrieve the list of files from a specified directory on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0–Success 1– Error

Restrictions	N/A
Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see <i>SFTP Client Adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client LIST service:

Field	Description
RemoteFileName	File name or pattern to do the listing for. If not included, the listing is for a * pattern (depending on server configuration). Optional. Only one * is allowed. For example, the following are valid: <ul style="list-style-type: none"> ◆ RemoteFileName=text* . ◆ RemoteFileName=*text The following is not valid: <ul style="list-style-type: none"> ◆ RemoteFileName=*text* The following returns all files, including parent directory and directory: <ul style="list-style-type: none"> ◆ *
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is ResponseTimeout specified in SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client LIST service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none">◆ OK - 0◆ No Such File - 2◆ Permission Denied - 3◆ General Failure - 4◆ Bad Message - 5◆ No Connection - 6◆ Connection Lost - 7◆ Operation Unsupported - 8
Files	Provides information about the files included in the specified directory.
Name	The name of the file or message.
Size	The size of the file or message.
Type	The type of the file or message. Most common values are: <ul style="list-style-type: none">◆ Regular◆ Directory
Permissions	A formatted permissions string. When getting a listing from a Application SFTP server, messages have permissions equal to "-rw-----", mailboxes have permissions equal to "-drwx-----".
ModificationTime	The date/time the file was last modified.
Owner	When getting a listing from a Application SFTP server, messages have an owner identity equal to "200", mailboxes have an owner identity equal to "300".
Group	When getting a listing from a Application SFTP server, messages and mailboxes have a group identity equal to "100".

Business Process Example

The following example business process excerpt illustrates using the SFTP Client LIST service:

```
[[Insert Begin Session]]
```

```
<operation name="SFTP Client LIST Service">  
  <participant name="SFTPClientList"/>  
  <output message="SFTPClientListServiceTypeInputMessage">
```

```
    <assign to="SessionToken"
from="/SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="." from="*"></assign>
</output>
<input message="inmsg">
    <assign to="SFTPClientListServiceResults" from="*"></assign>
</input>
</operation>

[[Insert End Session]]
```

SFTP Client LIST Service (Build 4325 or higher)

The following table provides an overview of the SFTP Client LIST service:

System name	SFTP Client LIST Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Retrieves a list of files on a specified directory on the trading partner's SFTP server.
Business usage	Use this service to retrieve a list of files in a specified directory on the trading partner's system and return the list to Application when the SFTP protocol is required as the transport mechanism.
Usage example	A business process is executed to retrieve a list of files from the external trading partner. Application uses the SFTP Client LIST service, working through the SFTP Client adapter, to retrieve the list of files from a specified directory on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0–Success 1– Error

Restrictions	N/A
Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see <i>SFTP Client Adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client LIST service:

Field	Description
RemoteFileName	File name or pattern to do the listing for. If not included, the listing is for a * pattern (depending on server configuration). Optional. Only one * is allowed. For example, the following are valid: <ul style="list-style-type: none"> ◆ RemoteFileName=text* . ◆ RemoteFileName=*text The following is not valid: <ul style="list-style-type: none"> ◆ RemoteFileName=*text* The following returns all files, including parent directory and directory: <ul style="list-style-type: none"> ◆ *
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is ResponseTimeout specified in SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client LIST service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none">◆ OK - 0◆ No Such File - 2◆ Permission Denied - 3◆ General Failure - 4◆ Bad Message - 5◆ No Connection - 6◆ Connection Lost - 7◆ Operation Unsupported - 8
Files	Provides information about the files included in the specified directory.
Name	The name of the file or message.
Size	The size of the file or message.
Type	The type of the file or message. Most common values are: <ul style="list-style-type: none">◆ Regular◆ Directory
Permissions	A formatted permissions string. When getting a listing from a Application SFTP server, messages have permissions equal to "-rw-----", mailboxes have permissions equal to "-drwx-----".
ModificationTime	The date/time the file was last modified.
Owner	When getting a listing from a Application SFTP server, messages have an owner identity equal to "200", mailboxes have an owner identity equal to "300".
Group	When getting a listing from a Application SFTP server, messages and mailboxes have a group identity equal to "100".

Business Process Example

The following example business process excerpt illustrates using the SFTP Client LIST service:

```
[[Insert Begin Session]]
```

```
<operation name="SFTP Client LIST Service">  
  <participant name="SFTPClientList"/>  
  <output message="SFTPClientListServiceTypeInputMessage">
```

```
    <assign to="SessionToken"
from="/SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="." from="*"></assign>
</output>
<input message="inmsg">
    <assign to="SFTPClientListServiceResults" from="*"></assign>
</input>
</operation>

[[Insert End Session]]
```

SFTP Client MOVE Service

The following table provides an overview of the SFTP Client MOVE service:

System name	SFTP Client MOVE Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Moves or renames a document from one directory to another on the trading partner's SFTP server.
Business usage	Use this service to move or rename a document on the trading partner system when the SFTP protocol is required as the communication mechanism with the trading partner.
Usage example	A business process is executed to move or rename a document on the trading partner's system. Application uses the SFTP Client MOVE service, working through the SFTP Client adapter, to move the specified document from one directory to another on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0–Success 1– Error

Restrictions	N/A
Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see the <i>SFTP Client adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client MOVE service:

Field	Description
RemoteFromFileName	The current name of the remote file. Required. Note: Wildcards are not supported.
RemoteToFileName	The new name of the remote file. Required.
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is ResponseTimeout specified in SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client MOVE service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none"> ◆ OK - 0 ◆ No Such File - 2 ◆ Permission Denied - 3 ◆ General Failure - 4 ◆ Bad Message - 5 ◆ No Connection - 6 ◆ Connection Lost - 7 ◆ Operation Unsupported - 8

Business Process Example

The following example business process excerpt illustrates using the SFTP Client MOVE service:

```
[[Insert Begin Session]]

<operation name="SFTP MOVE SERVICE">
  <participant name="SFTPClientMove"/>
  <output message="MoveRequest1">
    <assign to="SessionToken"
from="/ProcessData/SftpBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="RemoteFromFileName">OldFilename</assign>
    <assign to="RemoteToFileName">NewFilename</assign>
  </output>
  <input message="inmsg">
    <assign to="SftpMoveResults" from="*"></assign>
  </input>
</operation>

[[Insert End Session]]
```

SFTP Client PUT Service

The following table provides an overview of the SFTP Client PUT service:

System name	SFTP Client PUT Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Places a document in a specified directory on the trading partner's SFTP server.
Business usage	Use this service to transfer a document from Application to a trading partner when the SFTP protocol is required as the transport mechanism.
Usage example	A business process is executed that translates a document to send to a trading partner. After the translation, Application uses the SFTP Client PUT service, working through the SFTP Client adapter, to place the document in a specified directory on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	<p>The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations:</p> <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service <p>The SFTP Client PUT service must be placed between an SFTP Begin Session service and an SFTP End Session service. It may be used to put a document that is returned from an SFTP Client GET service.</p>
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None

Returned status values	0–Success 1– Error
Restrictions	N/A
Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see the <i>SFTP Client adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client PUT service:

Field	Description
DocumentId	The identifier of a document to be copied to the remote server. If not specified, the primary document will be copied. If not specified and there is not a primary document, an error will be reported. Optional.
RemoteFileName	The name of the file used to place the document on the remote trading partner server. If not specified, the name of the document will be used. Optional.
ResponseTimeout	The maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is the ResponseTimeout value specified in the SFTP Client Begin Session service.
SessionToken	The returned SessionToken from the Begin Session service. Required.
Primary Document	The file transferred as a result of the PUT service.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client PUT service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none">◆ OK - 0◆ No Such File - 2◆ Permission Denied - 3◆ General Failure - 4◆ Bad Message - 5◆ No Connection - 6◆ Connection Lost - 7◆ Operation Unsupported - 8

Business Process Example

The following business process excerpt uses the SFTP Client Adapter to send the primary document from Application to the remote SFTP server using the SFTP Client PUT service:

```
[[Insert Begin Session]]
```

```
<operation name="SFTP PUT SERVICE">
  <participant name="SFTPClientPut"/>
  <output message="PutRequest">
    <assign to="SessionToken"
from="/ProcessData/SftpBeginSessionServiceResults/SessionToken/text()"></assign>
    <assign to="RemoteFileName">FilenameToPut</assign>
    <assign to="." From="PrimaryDocument"></assign>
  </output>
  <input message="inmsg">
    <assign to="SftpPutServiceResults" from="*"></assign>
  </input>
</operation>
```

```
[[Insert End Session]]
```

The following business process excerpt uses the SFTP Client Adapter to send a document received from a GET from Application to the remote SFTP server:

```
[[Insert Begin Session]]
```

```
<operation name="Get">
  <participant name="SFTPClientGet"/>
  <output message="GetRequest">
    <assign to="SessionToken"
from="/ProcessData/SftpBeginSessionResults/SessionToken/text()"></assign>
```

```

    <assign to="RemoteFileName">FilenameToGet</assign>
  </output>
  <input message="GetResults">
    <assign to="GetResults" from="DocumentId"/>
  </input>
</operation>

<operation name="Put">
  <participant name="SFTPClientPut"/>
  <output message="PutRequest">
    <assign to="SessionToken"
from="/ProcessData/SftpBeginSessionResults/SessionToken/text()"></assign>
    <assign to="." From="/ProcessData/GetResults/DocumentId"/>
  <input message="SftpPutResults">
    <assign to="PutResults" from="*"></assign>
  </input>
</operation>

[[Insert End Session]]

```

Correlations and Document Tracking

The following table details the correlations available from the SFTP Client PUT service for document tracking:

Key	Values
ACTION	Get, Put
Direction	Inbound, Outbound
Protocol	SFTP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost
Username	username
RemoteFile	filename

SFTP Client PWD Service

The following table provides an overview of the SFTP Client PWD service:

System name	SFTP Client PWD Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Retrieves the present working directory from the trading partner's SFTP server.
Business usage	Use this service to get information about the current working directory on the trading partner SFTP server.
Usage example	A business process is executed to place a document in a specific directory on the trading partner's system. To include the name of the directory where the document was placed, use the SFTP Client PWD service to place the directory information in the business process.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: <ul style="list-style-type: none">◆ SFTP Client Begin Session service◆ SFTP Client CD service◆ SFTP Client DELETE service◆ SFTP Client End Session service◆ SFTP Client GET service◆ SFTP Client LIST service◆ SFTP Client MOVE service◆ SFTP Client PUT service◆ SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0–Success 1– Error
Restrictions	N/A

Persistence level	System Default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see the <i>SFTP Client adapter</i> . Debug information for this service is located at: Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client PWD service:

Field	Description
SessionToken	Specifies the identifier for the session established between the SFTP Client adapter and an SFTP server. Required.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client PWD service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: <ul style="list-style-type: none"> ◆ OK - 0 ◆ No Such File - 2 ◆ Permission Denied - 3 ◆ General Failure - 4 ◆ Bad Message - 5 ◆ No Connection - 6 ◆ Connection Lost - 7 ◆ Operation Unsupported - 8
Pwd Directory	Specifies the current directory.

Business Process Example

The following example business process excerpt illustrates using the SFTP Client PWD service:

```
[[Insert Begin Session]]
```

```
<operation name="SFTP PWD SERVICE">
```

```
<participant name="SFTPClientPwd" />
  <output message="PwdRequest">
    <assign to="SessionToken"
from="/ProcessData/sftpBeginSessionServiceResults/SessionToken/text()"></assign>
  </output>
  <input message="inmsg">
    <assign to="SFTPClientPwdResults" from="*"></assign>
  </input>
</operation>
</sequence>

[[Insert End Session]]
```

SFTP Server Adapter (Build 4300 - Build 4323)

The SFTP Server adapter has the following major features:

- Uses perimeter services
- Uses Mailbox subsystem as its repository (virtual roots)
- Routing rules for items placed in Mailbox can be used to trigger a business process
- Supports SSH2 with SFTP version 3 or lower
- Supports inbound SSH/SFTP and SSH/SCP protocols

The following table provides an overview of the SFTP Server adapter:

System name	SFTP Server Adapter
Graphical Process Modeler (GPM) category	None
Description	Receives and processes requests from external trading partners that are submitted through the SFTP protocol or SCP protocol.
Business usage	Use this adapter to enable external SFTP clients or SCP clients to <i>put</i> files into a Application Mailbox or <i>get</i> files from a Application Mailbox.
Usage example	A trading partner uses an SFTP client to retrieve a business document from his Application Mailbox.
Preconfigured?	DemoAllSFTPServerAdapter is fully preconfigured and enabled when you perform the demo procedure. See <i>Run SFTPClientDemoAllServices</i> . SFTP Server Adapter is partially preconfigured. Because both configurations specify the same port, only one of these adapters can be enabled at a time. DemoAllSFTPServerAdapter is enabled after installation of Application. To enable SFTP Server Adapter, you must first disable DemoAllSFTPServerAdapter or change the port assignment.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	Perimeter services
Application requirements	An SFTP or SCP client at the external trading partner location. When this adapter is configured with a 'non-local-mode' perimeter server, the perimeter server must be installed and running. The perimeter server is typically installed in a DMZ environment, separated from Application by a firewall. Refer to the perimeter services documentation for details on installing and running that component. If users exceed a maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	No

Invocation	This adapter is not invoked from a business process.
Business process context considerations	None
Returned status values	N/A
Restrictions	<p>Restricted to platforms that support Java SDK version 1.4 and above.</p> <p>Transfer resumption is disabled by default. To enable transfer resumption and listing documents that are in the staging area, edit the <code>sftp.properties</code> file (located at <code><install_dir>/properties/sftp.properties.in</code>) to set <code>listStagedDocuments = True</code> (default is <code>False</code>).</p> <p>To support transfer resumption, the SFTP Server Adapter keeps partial documents in a temporary document staging area. This allows SFTP clients to resume a transfer (within a specified timeframe). If the transfer does not resume within the specified amount of time, the Partial Document Clean Up Service removes documents from the staging area and the transfer is no longer available for resumption.</p> <p>A common behavior among SFTP clients before resuming a transfer is to request a list of the directory contents. In response to list requests, the default behavior is for the SFTP Server Adapter to return a listing that includes (1) complete documents in the target mailbox and (2) partial documents in the staging area.</p> <p>Note: Partial documents are assigned to a particular user. The system only displays partial documents to the user to whom they are assigned.</p> <p>If two documents with the same name exist in both the mailbox and the document staging area, only the partial document in the staging area is displayed in response to a list request.</p> <p>You can change the default behavior by editing the <code>sftp.properties</code> file. To enable listing documents that are in the staging area, set <code>listStagedDocuments = True</code>. Default is <code>False</code>.</p> <p>The SFTP Server adapter does not return nonextractable files as part of a directory listing. Once a message becomes nonextractable, it effectively disappears from the SFTP view of the mailbox.</p> <p>The home directory for SFTP is a virtual root mailbox in Application. The mailbox can include both extractable and nonextractable messages. When the SFTP Server adapter accesses the mailbox, only extractable messages are displayed.</p>

Permissions	<p>To access the SFTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or by default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root and full rights. Rights that can be given on behalf of a user are: write, read, execute, view, and delete. Each right allows specific actions to be performed. By default, a user assigned to a mailbox has all available rights.</p> <p>If a user needs to fully operate on a mailbox at a lower level in the mailbox hierarchy, the user must also have permission and rights on all mailboxes that are between the target mailbox and his virtual root. Rights required for mailbox operations are:</p> <ul style="list-style-type: none"> ◆ Add a message to a mailbox - Write permission for the Mailbox ◆ Extract message from mailbox - Read for the Mailbox ◆ List submailbox - Execute for All mailboxes from virtual root to submailbox ◆ List virtual root mailbox - Execute for the Virtual root mailbox ◆ List virtual root mailbox without mailbox execute permission - Execute for the MailboxLoginWithoutVirtualRootPermission ◆ Login if ACL active - Execute for Server Permission ◆ Login to the virtual root mailbox - Execute for Virtual root mailbox ◆ Login to the virtual root mailbox without mailbox execute permission - Execute for MailboxLoginWithoutVirtualRootPermission ◆ Move message to mailbox - Write for Destination Mailbox ◆ Remove message from mailbox Delete Mailbox <p>Restricted operation can be granted to users with a permission named <i>MailboxLoginWithoutVirtualRootPermission</i>. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes.</p>
Persistence level	Default
Testing considerations	<p>At Application startup, attempt to access SFTP server using a supported SFTP client with the configured IP/Port.</p> <p>Debug information can be found in the SFTP logs. Select Logging Level from the following:</p> <ul style="list-style-type: none"> ◆ Error - only errors. ◆ Communication Trace - errors, requests from clients, and responses from the Server adapter. This includes ACL violations. ◆ All - for debugging, all activities.

Implementing the SFTP Server Adapter

To implement the SFTP Server adapter, complete the following tasks:

1. Create a configuration of the SFTP Server adapter (or enable the configuration installed with Application and edit parameters as needed).
2. Configure the SFTP Server adapter.

Configuring the SFTP Server Adapter

To configure the SFTP Server adapter:

1. Select **Deployment > Services > Configuration**.
2. Next to New Service, click **Go!**
3. Select the List View icon, then select the **SFTP Server adapter** from the list. Click **Save**.
4. Click **Next**.
5. Specify field settings:

Field	Description
Name	Name this adapter will have in Application
Description	Description of adapter
Select a Group	None – Do not include this configuration in a group.
Perimeter Server	List of perimeter servers, including local-mode perimeter servers. Required. Default is Node 1 & Local.
Enabled Protocols	Select the protocols to enable for this adapter. Required. Valid values are: <ul style="list-style-type: none">◆ SFTP and SCP◆ SFTP◆ SCP Default is SFTP and SCP. Note: The SCP option is only available for new configurations of the SFTP Server adapter. If you have a previous version, you can disable it and create a new one to enable SCP or SFTP and SCP.
Host Identity Key	Private/Public key pair used to identify the Application SFTP Server to remote clients. Required.
SFTP Server Listen Port	The unique port number that the SFTP Server should bind to and listen on for connection requests. Cannot be used by any other adapter. Required.
Minimum Number of Threads	A tuning parameter that indicates the minimum number of threads that the perimeter server will use to improve performance. Optional. Default is 3. Note: Retain the default value unless instructed otherwise by Sterling Commerce Support.
Maximum Number of Threads	A tuning parameter that indicates the maximum number of threads that the perimeter server will use to improve performance. Optional. Default is 6. Note: Retain the default value unless instructed otherwise by Sterling Commerce Support.
Transfer Thread Pool Size	A tuning parameter that indicates the number of permanent transfer threads the server begins with. Once a socket has either been accepted or connected, the socket is registered with a transfer thread. This thread asynchronously performs all the input and output for the socket. If all the permanent threads become fully loaded, additional threads are created to handle additional connections and shut down once they have no sockets to service. Optional. Default is 2.

Field	Description
Channels per Transfer Thread	A tuning parameter that indicates the number of channels available for each transfer thread. Set maximum number of SelectableChannels that can be assigned to the accept, transfer, and connect selectors. Value of 1 effectively makes server behave in thread-per-connection mode. Optional. Default is 400.
Maximum Authentications	The maximum number of failed authentication attempts a user is allowed before the session is ended. Optional. Default is 3.
Session Timeout (seconds)	The number of seconds each session is allowed to last. Required. Valid value is any number between 1 and 9,999,999. Default is 120,000. Note: If the timeout is reached during a transfer, the session will be closed immediately after the transfer completes.
Resumption Timeout (hours)	Timeout value for the incomplete document before it is purged. Required. Valid value is any number between 1 and 9,999,999. Default is 48.
Compression	Specifies whether data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values: None, ZLIB.
PreferredCipher	The cipher the server prefers to use for both client to server and server to client stream encryption. Optional. Default is blowfish-cbc. Valid values are: <ul style="list-style-type: none"> ◆ 3des-cbc ◆ blowfish-cbc ◆ aes256-cbc ◆ aes192-cbc ◆ aes128-cbc ◆ cast128-cbc ◆ twofish256-cbc ◆ twofish192-cbc ◆ twofish128-cbc
PreferredMAC	The MAC the server prefers to use for stream encryption. Optional. Valid values are: <ul style="list-style-type: none"> ◆ hmac-sha1 ◆ hmac-md5 Default is hmac-sha1
Required Authentication	Specifies the type of authentication required for the adapter. Required. Valid values are: <ul style="list-style-type: none"> ◆ Password or Public Key (default) ◆ Password ◆ Public Key ◆ Password and Public Key

Field	Description
Maximum Logins	Maximum number of logins the adapter may have active at any point of time. Use this to limit the total number of users allowed to access a server at any one time. This can be used to manage server performance. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999.
Maximum Logins Per User	Maximum number of logins each user may have active on this adapter at any point of time. Use this to limit users who want to make many connections at the same time to ensure bandwidth is shared among users. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999.
Data Storage Type	Select whether documents will be stored on the file system, the database, or the system default. Required. Valid values are: <ul style="list-style-type: none"> ◆ File System (default) ◆ Database ◆ System Default
Should the adapter be restricted to a certain group of users?	Select Yes or No to indicate whether to restrict specific users and groups to access the SFTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages.
Users	Select a list of users who are granted permission to access the server.
Groups	Select a list of groups who are granted permission to access the server.
Extractable Count	The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Optional. Valid value is any integer.
Extractable For	A counter indicating the length of time (in days, hours and minutes) the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable Count. Optional. Format is dddhmm.
Extractable	A yes or no value indicating if this message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Optional.

6. On the Confirm screen, ensure that **Enable service for Business Process** is selected. Click **Finish**.

Correlations and Document Tracking

The following table details the correlations available from the SFTP Server service for document tracking:

Key	Values
ACTION	Get, Put
Direction	Inbound, Outbound
Protocol	SFTP or SCP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost

Key	Values
Username	username

Activity Monitoring for the SFTP Server Adapter

The SFTP Server adapter creates activity monitoring records for the following activities:

- Active sessions (connections to clients)

- In progress PUTs display the data transferred in kbps with a progress indicator

- In progress GETs display the data transferred in kbps

To view the records, select **Business Processes > Current Activities > SFTP Server Adapter**.

SFTP Server Adapter (Build 4324 or higher)

The SFTP Server adapter has the following major features:

- Uses perimeter services
- Uses Mailbox subsystem as its repository (virtual roots)
- Routing rules for items placed in Mailbox can be used to trigger a business process
- Supports SSH2 with SFTP version 3 or lower
- Supports inbound SSH/SFTP and SSH/SCP protocols

The following table provides an overview of the SFTP Server adapter:

System name	SFTP Server Adapter
Graphical Process Modeler (GPM) category	None
Description	Receives and processes requests from external trading partners that are submitted through the SFTP protocol or SCP protocol.
Business usage	Use this adapter to enable external SFTP clients or SCP clients to <i>put</i> files into a Application Mailbox or <i>get</i> files from a Application Mailbox.
Usage example	A trading partner uses an SFTP client to retrieve a business document from his Application Mailbox.
Preconfigured?	DemoAllSFTPServerAdapter is fully preconfigured and enabled when you perform the demo procedure. See <i>Run SFTPClientDemoAllServices</i> . SFTP Server Adapter is partially preconfigured. Because both configurations specify the same port, only one of these adapters can be enabled at a time. DemoAllSFTPServerAdapter is enabled after installation of Application. To enable SFTP Server Adapter, you must first disable DemoAllSFTPServerAdapter or change the port assignment.
Requires third party files?	No
Platform availability	All Application supported platforms
Related services	Perimeter services
Application requirements	An SFTP or SCP client at the external trading partner location. When this adapter is configured with a 'non-local-mode' perimeter server, the perimeter server must be installed and running. The perimeter server is typically installed in a DMZ environment, separated from Application by a firewall. Refer to the perimeter services documentation for details on installing and running that component. If users exceed a maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	No

Invocation	This adapter is not invoked from a business process.
Business process context considerations	None
Returned status values	N/A
Restrictions	<p>Restricted to platforms that support Java SDK version 1.4 and above.</p> <p>Transfer resumption is disabled by default. To enable transfer resumption and listing documents that are in the staging area, edit the <code>sftp.properties</code> file (located at <code><install_dir>/properties/sftp.properties.in</code>) to set <code>listStagedDocuments = True</code> (default is <code>False</code>).</p> <p>To support transfer resumption, the SFTP Server Adapter keeps partial documents in a temporary document staging area. This allows SFTP clients to resume a transfer (within a specified timeframe). If the transfer does not resume within the specified amount of time, the Partial Document Clean Up Service removes documents from the staging area and the transfer is no longer available for resumption.</p> <p>A common behavior among SFTP clients before resuming a transfer is to request a list of the directory contents. In response to list requests, the default behavior is for the SFTP Server Adapter to return a listing that includes (1) complete documents in the target mailbox and (2) partial documents in the staging area.</p> <p>Note: Partial documents are assigned to a particular user. The system only displays partial documents to the user to whom they are assigned.</p> <p>If two documents with the same name exist in both the mailbox and the document staging area, only the partial document in the staging area is displayed in response to a list request.</p> <p>You can change the default behavior by editing the <code>sftp.properties</code> file. To enable listing documents that are in the staging area, set <code>listStagedDocuments = True</code>. Default is <code>False</code>.</p> <p>The SFTP Server adapter does not return nonextractable files as part of a directory listing. Once a message becomes nonextractable, it effectively disappears from the SFTP view of the mailbox.</p> <p>The home directory for SFTP is a virtual root mailbox in Application. The mailbox can include both extractable and nonextractable messages. When the SFTP Server adapter accesses the mailbox, only extractable messages are displayed.</p>

Permissions	<p>To access the SFTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or by default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root and full rights. Rights that can be given on behalf of a user are: write, read, execute, view, and delete. Each right allows specific actions to be performed. By default, a user assigned to a mailbox has all available rights.</p> <p>If a user needs to fully operate on a mailbox at a lower level in the mailbox hierarchy, the user must also have permission and rights on all mailboxes that are between the target mailbox and his virtual root. Rights required for mailbox operations are:</p> <ul style="list-style-type: none"> ◆ Add a message to a mailbox - Write permission for the Mailbox ◆ Extract message from mailbox - Read for the Mailbox ◆ List submailbox - Execute for All mailboxes from virtual root to submailbox ◆ List virtual root mailbox - Execute for the Virtual root mailbox ◆ List virtual root mailbox without mailbox execute permission - Execute for the MailboxLoginWithoutVirtualRootPermission ◆ Login if ACL active - Execute for Server Permission ◆ Login to the virtual root mailbox - Execute for Virtual root mailbox ◆ Login to the virtual root mailbox without mailbox execute permission - Execute for MailboxLoginWithoutVirtualRootPermission ◆ Move message to mailbox - Write for Destination Mailbox ◆ Remove message from mailbox Delete Mailbox <p>Restricted operation can be granted to users with a permission named <i>MailboxLoginWithoutVirtualRootPermission</i>. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes.</p>
Persistence level	Default
Testing considerations	<p>At Application startup, attempt to access SFTP server using a supported SFTP client with the configured IP/Port.</p> <p>Debug information can be found in the SFTP logs. Select Logging Level from the following:</p> <ul style="list-style-type: none"> ◆ Error - only errors. ◆ Communication Trace - errors, requests from clients, and responses from the Server adapter. This includes ACL violations. ◆ All - for debugging, all activities.

Implementing the SFTP Server Adapter

To implement the SFTP Server adapter, complete the following tasks:

1. Create a configuration of the SFTP Server adapter (or enable the configuration installed with Application and edit parameters as needed).
2. Configure the SFTP Server adapter.

Configuring the SFTP Server Adapter

To configure the SFTP Server adapter:

1. Select **Deployment > Services > Configuration**.
2. Next to New Service, click **Go!**
3. Select the List View icon, then select the **SFTP Server adapter** from the list. Click **Save**.
4. Click **Next**.
5. Specify field settings:

Field	Description
Name	Name this adapter will have in Application
Description	Description of adapter
Select a Group	None – Do not include this configuration in a group.
Perimeter Server	List of perimeter servers, including local-mode perimeter servers. Required. Default is Node 1 & Local.
Enabled Protocols	Select the protocols to enable for this adapter. Required. Valid values are: <ul style="list-style-type: none">◆ SFTP and SCP◆ SFTP◆ SCP Default is SFTP and SCP. Note: The SCP option is only available for new configurations of the SFTP Server adapter. If you have a previous version, you can disable it and create a new one to enable SCP or SFTP and SCP.
Host Identity Key	Private/Public key pair used to identify the Application SFTP Server to remote clients. Required.
SFTP Server Listen Port	The unique port number that the SFTP Server should bind to and listen on for connection requests. Cannot be used by any other adapter. Required.
Minimum Number of Threads	A tuning parameter that indicates the minimum number of threads that the perimeter server will use to improve performance. Optional. Default is 3. Note: Retain the default value unless instructed otherwise by Sterling Commerce Support.
Maximum Number of Threads	A tuning parameter that indicates the maximum number of threads that the perimeter server will use to improve performance. Optional. Default is 6. Note: Retain the default value unless instructed otherwise by Sterling Commerce Support.
Transfer Thread Pool Size	A tuning parameter that indicates the number of permanent transfer threads the server begins with. Once a socket has either been accepted or connected, the socket is registered with a transfer thread. This thread asynchronously performs all the input and output for the socket. If all the permanent threads become fully loaded, additional threads are created to handle additional connections and shut down once they have no sockets to service. Optional. Default is 2.

Field	Description
Channels per Transfer Thread	A tuning parameter that indicates the number of channels available for each transfer thread. Set maximum number of SelectableChannels that can be assigned to the accept, transfer, and connect selectors. Value of 1 effectively makes server behave in thread-per-connection mode. Optional. Default is 400.
Maximum Authentications	The maximum number of failed authentication attempts a user is allowed before the session is ended. Optional. Default is 3.
Session Timeout (seconds)	The number of seconds each session is allowed to last. Required. Valid value is any number between 1 and 9,999,999. Default is 120,000. Note: If the timeout is reached during a transfer, the session will be closed immediately after the transfer completes.
Resumption Timeout (hours)	Timeout value for the incomplete document before it is purged. Required. Valid value is any number between 1 and 9,999,999. Default is 48.
Compression	Specifies whether data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values: None, ZLIB.
PreferredCipher	The cipher the server prefers to use for both client to server and server to client stream encryption. Optional. Default is blowfish-cbc. Valid values are: <ul style="list-style-type: none"> ◆ 3des-cbc ◆ blowfish-cbc ◆ aes256-cbc ◆ aes192-cbc ◆ aes128-cbc ◆ cast128-cbc ◆ twofish256-cbc ◆ twofish192-cbc ◆ twofish128-cbc
PreferredMAC	The MAC the server prefers to use for stream encryption. Optional. Valid values are: <ul style="list-style-type: none"> ◆ hmac-sha1 ◆ hmac-md5 Default is hmac-sha1
Required Authentication	Specifies the type of authentication required for the adapter. Required. Valid values are: <ul style="list-style-type: none"> ◆ Password or Public Key (default) ◆ Password ◆ Public Key ◆ Password and Public Key

Field	Description
Maximum Logins	Maximum number of logins the adapter may have active at any point of time. Use this to limit the total number of users allowed to access a server at any one time. This can be used to manage server performance. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999.
Maximum Logins Per User	Maximum number of logins each user may have active on this adapter at any point of time. Use this to limit users who want to make many connections at the same time to ensure bandwidth is shared among users. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999.
Data Storage Type	Select whether documents will be stored on the file system, the database, or the system default. Required. Valid values are: <ul style="list-style-type: none"> ◆ File System (default) ◆ Database ◆ System Default
Should the adapter be restricted to a certain group of users?	Select Yes or No to indicate whether to restrict specific users and groups to access the SFTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages.
Users	Select a list of users who are granted permission to access the server.
Groups	Select a list of groups who are granted permission to access the server.
Extractable Count	The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Optional. Valid value is any integer.
Extractable For	A counter indicating the length of time (in days, hours and minutes) the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable Count. Optional. Format is dddhmm.
Extractable	A yes or no value indicating if this message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Optional.

6. On the Confirm screen, ensure that **Enable service for Business Process** is selected. Click **Finish**.

Correlations and Document Tracking

The following table details the correlations available from the SFTP Server service for document tracking:

Key	Values
ACTION	Get, Put
Direction	Inbound, Outbound
Protocol	SFTP or SCP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost

Key	Values
Username	username

Activity Monitoring for the SFTP Server Adapter

The SFTP Server adapter creates activity monitoring records for the following activities:

- Active sessions (connections to clients)

- In progress PUTs display the data transferred in kbps with a progress indicator

- In progress GETs display the data transferred in kbps

To view the records, select **Business Processes > Current Activities > SFTP Server Adapter**.

Simple Network Management Protocol (SNMP) Trap Adapter

The following table provides an overview of the SNMP Trap adapter:

System name	None
Graphical Process Modeler (GPM) category	None
Description	<p>You can use the SNMP Trap adapter in either of the following two ways:</p> <ul style="list-style-type: none">◆ SNMP trap generator – The SNMP Trap adapter used as a trap generator enables business processes to send notifications through SNMPv1 traps to SNMP management stations about business process progress or faults. For more information, see <i>How the SNMP Trap Adapter Works as a Trap Generator</i> on page 1438.◆ SNMP trap receiver – The SNMP Trap adapter used as a trap receiver, while listening on a specified port, starts a business process upon receipt of an incoming SNMPv1 trap. For more information, see <i>How the SNMP Trap Adapter Works As a Trap Receiver</i> on page 1441.
Preconfigured?	No
Requires third party files?	Slv1_snmp4_10.jar
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Restrictions	See <i>Requirements</i> on page 1437.

Requirements

To set up and use the SNMP Trap adapter, you should understand SNMP concepts, such as how to generate and receive traps, especially enterprise trap definitions.

The SNMP Trap adapter has the following limitations:

- Supports the receipt and generation of SNMPv1 traps only

- Does not support polling by an SNMP manager, because the SNMP Trap adapter is not a full SNMP agent

- Does not support SNMP Get or Set commands

If you want to generate a custom trap, you may need to create a Management Information Base (MIB) definition for the custom trap, if one is not already defined in the SNMP manager

When using the SNMP Trap adapter as a trap generator, there is no guarantee that the SNMPv1 trap will be received, and there is no way to confirm that the trap has been received

To verify that traps generated by the SNMP Trap adapter are being received, use the SNMP Trap adapter as a trap receiver

How the SNMP Trap Adapter Works as a Trap Generator

SNMP was designed to be an application-level protocol that is part of the TCP/IP suite and runs over the User Datagram Protocol (UDP). Because SNMP relies on UDP, which is a connectionless protocol, SNMP is connectionless. Therefore, no ongoing connections are maintained between a management station and its agents. Each exchange is a separate transaction.

Because UDP is connectionless, there is no guarantee of delivery. SNMPv1 does not acknowledge traps; the SNMP trap generator reports success to the business process even when the trap is not received by the SNMP management stations. This condition prevents a business process from halting in the event the SNMP trap generator fails.

Note: To increase the chances of delivery, you can configure the SNMP trap generator to send multiple traps to the same SNMP management station.

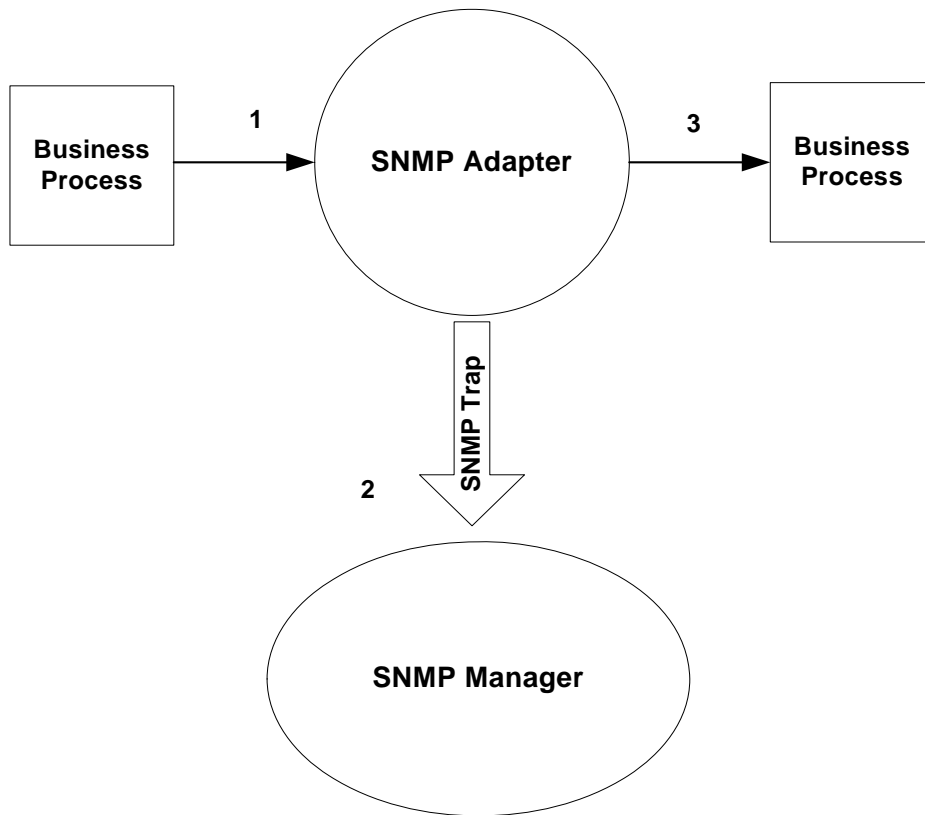
The SNMP trap generator does not require a document for input; the adapter obtains all inputs through business process parameters that are set either during adapter configuration, business process definition, or at run time.

The following steps summarize how the trap generator works:

1. The business process starts the trap generator, using the following business process parameters:
 - ◆ SNMP generic trap type
 - ◆ Specific trap code
 - ◆ Associated trap data
 - ◆ One or more SNMP management station addresses (IP address and port number)
2. The trap generator creates the trap and sends it to each SNMP management station address specified in step 1.
3. The trap generator returns the status of Success to the business process, regardless of whether the trap was generated and received successfully. When trap generation fails, the status report indicates the cause of the failure.

View the status report in Application by selecting **Business Process > Monitor > Current Processes** and then clicking the ID of the appropriate business process.

The following figure shows how the SNMP Trap adapter as a trap generator communicates with an SNMP manager or trap receiver during a business process:



Example

For example, you can use an SNMP management station to monitor your network. To monitor the progress of a business process running inside Application, configure a trap generator with your SNMP management station as a trap receiver. Configure the trap generator to send an SNMP trap message when specific events occur.

The following steps summarize how the trap generator works in the context of this example:

1. Configure the trap generator within Application.
You can also specify the needed information just before starting the business process. If you specify the information before starting the business process, you will override any values specified in the adapter configuration.
2. When the business process needs to inform an SNMP management station of an event, it starts the trap generator.
3. The trap generator takes the trap information, generates an SNMP trap message, and sends it to all the specified trap receivers.
4. The trap generator creates a status report and returns a status of Success to Application.

5. Application performs the next step in the business process.

Implementing the SNMP Trap Adapter as a Trap Generator

To implement the SNMP Trap adapter as a trap generator, complete the following tasks:

1. Create an SNMP Trap adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the SNMP Trap adapter. For information, see *Configuring the SNMP Trap Adapter as a Trap Generator* on page 1440.
3. Use the SNMP Trap adapter in a business process.

Configuring the SNMP Trap Adapter as a Trap Generator

When configuring the SNMP Trap adapter, you are defining the SNMP management stations to which the trap has to be sent and the type of trap.

Because this is a stateless adapter, you can use a single configuration of the adapter to send different types of traps to different management stations in multiple business processes.

When including trap-related information, use multiple configurations of the SNMP trap generator. The configuration of trap-related variables must include a named process data parameter that contains the value to be sent for that variable. This parameter can be different between business processes.

Caution: Because the SNMP Trap adapter always returns a status of Success, it can be easy to overlook errors in trap configuration. Remember to check the status report of the adapter to verify whether the trap generation was successful. The SNMP Trap adapter always generates a status report, even when the trap generation is successful. When trap generation fails, the status report indicates the cause of the failure.

To configure the SNMP Trap adapter, you must specify settings for the following fields in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.

Field	Description
Trap Receivers (trapReceivers)	<p>List of SNMP management stations to which the trap is sent. Specify as a comma- or semicolon-delimited list of host:port strings representing SNMP management stations. Specify the same address multiple times to send multiple trap messages to the same address. Required.</p> <p>For example, <i>IPAddress:Port</i>, where <i>IPAddress</i> is the IP address of the host and <i>Port</i> is the port number of the host on which the SNMP manager is running. The port is optional, unless the SNMP manager is running on a nonstandard port. The default value is 162.</p>
Community Name (community)	Access environment for a group of network management stations. If left blank, defaults to <i>public</i> .
Enterprise OID (enterpriseOID)	For generic traps, the object ID of the system generating the trap. For enterprise traps, the valid object ID of the management authority that defined the trap. Value is a valid object ID of an SNMP Management Information Base (MIB) object representing an enterprise. Defaults to the Sterling Commerce enterprise OID (1.3.6.1.4.1.1733).
Generic Trap Type (genericTrap)	<p>Type (value) of a generic SNMP trap. Required. Values are 0 - 5. Use 6 for enterprise or custom traps. The generic traps are fixed and new ones cannot be defined. The following is a list of the trap types and their corresponding values:</p> <ul style="list-style-type: none"> ◆ coldStart (0) ◆ warmStart (1) ◆ linkDown (2) ◆ linkUp (3) ◆ authenticationFailure (4) ◆ egpNeighborLoss (5)
Specific Trap Type (specificTrap)	Type code for an enterprise trap. Use this in conjunction with the enterprise OID to identify (or decode) the trap. You can define multiple enterprise traps. Required for enterprise traps. Value is a valid integer representing an enterprise trap. Value for generic traps is 0.
Variable OID # <i>n</i> (variableOID_# <i>n</i>)	Object ID of the trap variable in the MIB file, a valid MIB object ID. Specify the value through the parameter Process Data Param # <i>n</i> .
Variable Type # <i>n</i> (variableType_# <i>n</i>)	Type of MIB object, specified in variable OID # <i>n</i> . Values are NULL, Integer, OctetString, object ID. The SNMP Trap adapter supports up to six of these name-value pairs to enable passing of trap variables.
Process Data Param # <i>n</i> (procDataParam_# <i>n</i>)	Name of the process data parameter that contains the value for the trap variable. Values should be parseable into the type specified in Variable Type # <i>n</i> .

How the SNMP Trap Adapter Works As a Trap Receiver

The SNMPv1 trap receiver enables Application to receive an SNMPv1 trap and start a business process to handle the trap. The trap receiver converts all the information in the SNMPv1 trap message into business process parameters and passes these to the designated business process. The SNMP Trap adapter, working as a trap receiver, is inbound only and does not respond to the SNMP agent or trap generator.

The SNMP trap receiver is not started inside a business process; therefore it does not support input data from Application. The only input to the trap receiver is the incoming SNMPv1 trap message. The SNMP trap receiver sends no document to the business process that is running.

You can configure multiple configurations of the SNMP trap receiver to listen on different ports, and a different business process can be associated with each configuration. You can use the SNMP trap receiver to receive generic as well as custom traps. There is no limit on how many variable bindings the incoming trap can contain.

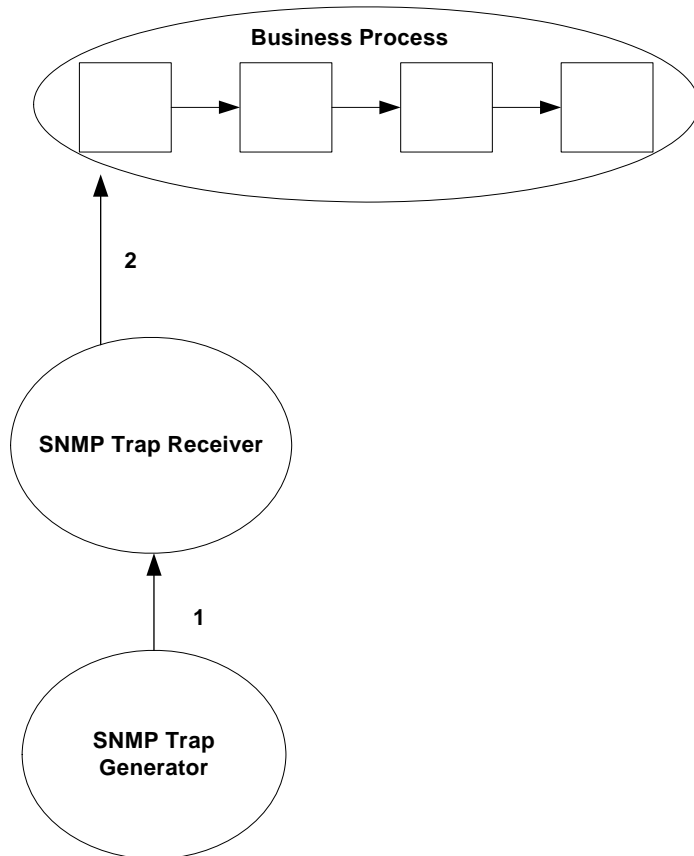
Note: Because the configuration for an SNMP trap receiver is not started within a business process, the configuration cannot be overridden.

The SNMP trap receiver operates over UDP; there is no guarantee that a trap will be received by Application. SNMPv1 traps are not acknowledged, so there is no confirmation message sent to the SNMP agent from which the trap was received.

The following steps summarize how the trap receiver works:

1. The SNMP trap receiver receives an incoming SNMPv1 trap.
2. The trap receiver passes trap parameters as business process parameters to Application and starts a related business process with the business process parameters for the trap.
3. The trap receiver continues to start business processes as it receives requests.

The following figure shows how the SNMP Trap adapter works as a trap receiver:



Example

You can use the SNMP trap receiver to receive SNMPv1 traps that have been generated from either an SNMP trap generator or an external SNMP manager. The SNMP trap generator in one installation of Application may generate a trap message to an SNMP Trap receiver in another installation of Application.

The following steps explain how the SNMP Trap adapter works as a trap receiver:

1. Configure the SNMP trap receiver.
2. The trap receiver starts listening for SNMPv1 traps on the specified port.
3. When receiving a trap message, the trap receiver checks the community name in the incoming trap message against any included/excluded communities specified in the trap receiver configuration.
4. After the filter rules have been applied, and the incoming trap message meets the criteria, the trap receiver starts the specified business process.

Business Process Variables

The following table contains process data variables for business processes started from the trap receiver. Multiple variables may be passed with any SNMP trap. The Name/Value process data variables always come in pairs.

Variable	Description
SNMP_ENTERPRISE_OID	Enterprise object ID. Unique ID for the enterprise.
SNMP_SPEC_TRAP_CODE	Unique trap ID for a specific Management Information Base (MIB). Can have any meaning that the enterprise specifies.
SNMP_GENERIC_TRAP_CODE	Predefined system trap.
SNMP_SENDING_IP	IP address of the trap sender.
SNMP_TIMESTAMP	Amount of time the trap sender has been running.
SNMP_RECEIVED_TIME	Time the trap was received.
VARIABLE_OID_n	Object ID of the nth variable passed with the trap.
VARIABLE_VALUE_n	Value of the nth variable passed with the trap.
SNMP_COMMUNITY_NAME	Community name in the incoming trap message.
VARIABLE_TYPE_n	Type of the nth variable passed with the trap (Octet String, Integer, Null, or Object ID)

Implementing the SNMP Trap Adapter As a Trap Receiver

To implement the SNMP Trap adapter as a trap receiver, complete the following tasks:

1. Create an SNMP Trap adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the SNMP Trap adapter. For information, see *Configuring the SNMP Trap Adapter As a Trap Receiver* on page 1444.
3. Use the SNMP Trap adapter in a business process.

Configuring the SNMP Trap Adapter As a Trap Receiver

To configure the SNMP Trap adapter, you must specify settings for the following fields in the Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Trap Receiver Port	Port number on which this configuration of the trap receiver listens for SNMPv1 traps.
Business Process Name	Business process name to be started upon receipt of the SNMPv1 trap.
Include Communities	Comma- or semicolon-separated list of valid SNMP community names. The trap receiver filters incoming SNMPv1 trap messages based on community names. Any incoming messages that do not match an entry in the list of valid community names are discarded. A wildcard of * indicates that all community names are to be included. All discarded trap messages are logged in the system log.
Exclude Communities	<p>Comma- or semicolon-separated list of valid SNMP community names. The trap receiver filters incoming SNMPv1 trap messages to exclude any community names on this list. Any incoming messages that match an entry in the list of excluded community names are discarded. A wildcard of * indicates that all community names are to be excluded. All discarded trap messages are logged in the system log.</p> <p>Optional; if left blank, no community names are excluded.</p>

Precedence Rules

You must specify a list of valid community names when configuring the trap receiver. If a community name contained within a trap message does not match the list of included names, or matches an excluded name, the designated business process does not start. The trap receiver logs a message containing the unauthorized community name and the associated trap message.

The following rules of precedence apply to the list of included and excluded community names:

Excluded community names take precedence over included names.

If the same name appears in both lists, incoming traps with this name are discarded.

If the included list is set to a wildcard and the excluded list has a community name, all community names are allowed except for the one in the excluded list.

SMTP Send Adapter

The SMTP Send adapter allows documents to be sent to any valid e-mail address using an accessible (SMTP-aware) mail server, usually the mail server of the trading partner that is running Application. It does this by enabling Application to mail (send) documents using SMTP, within Application, to the designated mail server. The back-end mail server system examines the document and does further processing, including the actual sending of the e-mail.

This adapter can be used when messages and documents need to be sent during the execution of a business process.

The following table provides an overview of the SMTP Send adapter:

System name	SMTP Send Adapter
Graphical Process Modeler (GPM) categories	All Services, Communications
Description	The Application SMTP Send adapter allows documents to be sent to any valid e-mail address by using an accessible mail server (usually of the trading partner that is running Application). It does this by enabling Application to mail (send) documents using SMTP, within Application, to the mail server.
Business usage	Documents and messages are sent by e-mail through the use of the SMTP Send adapter. Application mails (sends) documents to the mail server. The back-end mail server examines the document and does further processing. Note: For B2B applications and for transmitting EDI data, the required SMTP information can be specified in the trading profiles and accessed in a business process by using the B2B Send service. See <i>B2B Send System Service</i> .
Usage example	A business process runs with the trading partner information that sends/receives document(s). As part of the business process, the SMTP Send adapter picks up the primary document and sends it to the mail server, which does further processing of the document. This adapter is commonly used to send e-mail notification about documents that have been sent or processed or need some action from the recipient.
Preconfigured?	No
Requires third party files?	None
Platform availability	All supported Application platforms
Related Services	<ul style="list-style-type: none">◆ B2B Mail Client Adapter – Retrieves e-mail from a mail server. See <i>B2B Mail Client Adapter</i>.◆ XSLT Service – Useful for formatting readable e-mail body text. See <i>XSLT Service</i>.◆ MIME service – Packages MIME content e-mail messages. See <i>MIME Service</i>.
Application requirements	An internal SMTP mail server to send the e-mail.

Initiates business processes?	No. This adapter does not initiate business processes. This adapter cannot be used without a business process.
Invocation	Not applicable (internal service)
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the adapter in the WFC.
Returned status values	<ul style="list-style-type: none"> ◆ Success ◆ Service Configuration Error
Restrictions	<p>Three types of input document formats are supported by the SMTP Send adapter:</p> <ul style="list-style-type: none"> ◆ Free Text – Any formatted text input. The adapter constructs a MIME message of content type = text/plain and sends it to the mail server. ◆ Content of any RFC822-supported content type document – The content of the document is put into the primary document and the appropriate content type must be set in the corresponding document fields. ◆ Internet RFC822-compliant documents – An RFC822-compliant MIME multipart document is a MIME document having one or more attachments and has the sender and recipient addresses as well as the subject header written in the headers. The attachments can be of different file types (for example, Microsoft Word document, Excel spreadsheet). More information on the RFC822 standard can be found at http://www.faqs.org/rfcs/rfc822.html.
Testing considerations	<p>To test an instance of the SMTP Send adapter, create an SMTP_SEND_ADAPTER configuration. Set up a business process using this adapter to send an e-mail to a valid e-mail address. See <i>SMTP Send Adapter Business Process Usage</i> for setting up a business process.</p> <p>The most frequent problems encountered are:</p> <ul style="list-style-type: none"> ◆ Invalid host/port or invalid e-mail address ◆ Back-end mail server is not configured correctly ◆ Adapter is not active <p>If you get an error with any of these conditions, check with your system administrator (local and remote) to verify that the parameters you are using are correct.</p>

Note: The Application Mailbox services are separate services not associated with the SMTP Send adapter.

How the SMTP Send Adapter Works

The SMTP Send adapter can mail (send) documents to the back-end mail server of trading partners. The document is then e-mailed to the recipient designated in the To field of the configuration. This adapter picks up the document sent from the primary document in the business process.

The following section describes a business scenario and a sample solution using the SMTP Send adapter.

Business Scenario

A document has been sent to your trading partner for processing and you want to send an e-mail notifying them that the document has been sent and needs action. The message that you are sending is a standard text message telling the recipient that a document has been processed and what action is needed from them.

Business Solution Example

The following example shows a simple solution to this business scenario using the GPM. This example uses a File System adapter to pick up the text document containing the message to be sent. This document becomes the primary document. Since the `b2b-raw-message` value is set to false, this document is sent as the body of the e-mail.

Note: This document will be sent as a plain text document, so it must be either a text document or one that is readable as plain text. If the document contents are in XML, the XSLT service can be used to apply a style sheet to the XML to reformat it and make it more readable. See *XSLT Service*.

The message has the following properties:

FROM: sender@company1.com

TO: receiver@company2.com

SUBJECT: Document Notification

The mail host address for the mail server is 000.000.1.103.

The mail port for the mail server is 25.

GPM Example

A File System adapter has been configured to pick up a file that becomes the primary document used in the e-mail message.

The primary document is picked up by the SMTP adapter from the process data. The message is created with the parameters set by the SMTP adapter, and the primary document is the plain-text body of the message.

Name	Value	Use XPATH?
b2b-raw-message	false	<input type="checkbox"/>
smtpAuthenticationOpt		<input type="checkbox"/>
smtpEncodingType		<input type="checkbox"/>
smtpHost	myhost	<input type="checkbox"/>
smtpnotifyDelivery		<input type="checkbox"/>
smtpnotifyRead		<input type="checkbox"/>
smtpPassword		<input type="checkbox"/>
smtpPort	25	<input type="checkbox"/>
smtpRetries	3	<input type="checkbox"/>
smtpRetryInterval	1	<input type="checkbox"/>
smtpSubject		<input type="checkbox"/>
smtpUser		<input type="checkbox"/>
xport-smtp-auth		<input type="checkbox"/>
xport-smtp-mailBCC		<input type="checkbox"/>
xport-smtp-mailCC		<input type="checkbox"/>
xport-smtp-mailfrom	sender@company1.com	<input type="checkbox"/>
xport-smtp-mailhost	000.000.1.103	<input type="checkbox"/>
xport-smtp-mailport	25	<input type="checkbox"/>
xport-smtp-mailsubject	Document Notification	<input type="checkbox"/>
xport-smtp-mailsubjectencoding		<input type="checkbox"/>
xport-smtp-mailto	receiver@company2.com	<input type="checkbox"/>
xport-smtp-notify-delivery		<input type="checkbox"/>
xport-smtp-notify-read		<input type="checkbox"/>
xport-smtp-retries		<input type="checkbox"/>
xport-smtp-retryinterval		<input type="checkbox"/>
xport-tp-cakeycert-id		<input type="checkbox"/>
xport-tp-cipher		<input type="checkbox"/>
xport-tp-keycert-id		<input type="checkbox"/>
xport-tp-ssloption		<input type="checkbox"/>

Setting the b2b-raw-message to false defines the type of message contained in the primary document.

Business Process Modeling Language (BPML) Example

The following example shows the corresponding business process solution using BPML.

```
<process name="TestEmailExample1">
  <sequence name="Email Notification">
    <operation name="Pick Up Message">
      <participant name="MyFSA"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="appendOnExtract">>false</assign>
        <assign to="deleteAfterCollect">>false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="Send Email">
      <participant name="SMTP_SEND_ADAPTER"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
        <assign to="b2b-raw-message">>false</assign>
        <assign to="xport-smtp-mailfrom">sender@company1.com</assign>
        <assign to="xport-smtp-mailhost">000.000.0.000</assign>
        <assign to="xport-smtp-mailport">25</assign>
        <assign to="xport-smtp-mailsubject">Document Notification</assign>
        <assign to="xport-smtp-mailto">receiver@company2.com</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```



Implementing the SMTP Send Adapter

To implement the SMTP Send adapter for use in a business process:

1. Collect the following information:

The IP address or hostname of the mail server that you will be using to send your e-mail

The port for the mail server (usually this is 25)

Valid e-mail account(s) for the host mail server

If SMTP authorization is required to access the mail server

- ◆ Valid username
 - ◆ Valid password
2. Verify this information with the system administrator to make sure that the parameters are correct and that Application is able to access the mail server.
 3. Create a SMTP Send adapter configuration. For information, see *Managing Services and Adapters*.

4. Configure the SMTP Send adapter. For information, see *Configuring the SMTP Send Adapter* on page 1451.
5. Use the SMTP Send adapter in a business process.

Configuring the SMTP Send Adapter

To configure the SMTP Send adapter, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the SMTP Send adapter in Application:

Note: Most of the fields are optional when configuring an SMTP Send adapter. In most applications, it is only necessary to enter the name and description for the service configuration and set other fields in the business process.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Default SMTP Server	Trading partner SMTP mail server host system IP address or hostname. This is the host where the SMTP mail server is running – usually not the same as the Application mail server host. This is the SMTP mail server within the sending company. Optional. Valid values: <ul style="list-style-type: none"> ◆ Valid IP address ◆ Valid hostname Note: Can also be configured in the GPM.
Default SMTP Server Port	Port number for the Default SMTP Server. The port that the SMTP mail server listens on for incoming e-mail requests. Optional. Valid value is a valid port number. The default port for an SMTP service is 25. Note: Can also be configured in the GPM.
SMTP Authentication required	SMTP mail server requires authentication. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes – (true) Authenticates. ◆ No – (false) Does not authenticate. Default. Note: Can also be configured in the GPM.
User Name	User name for mail server authentication. Required if SMTP Authentication set to true. Valid value is a valid user name.

Field	Description
Password	Password for mail server authentication. Required if SMTP Authentication set to true. Valid value is a valid password.
Connection Retries	Number of times SMTP Send adapter tries to connect to the SMTP mail server if the connection is not successful. Optional. Valid value is an integer. Note: Can also be configured in the GPM.
Retry Interval (Sec)	Time delay between retrying the connection. Optional. Valid value is an integer. Note: Can also be configured in the GPM.
Subject	Mail message subject. This appears in the subject line of the e-mail. Optional. Note: Can also be configured in the GPM.
Encoding Type	The type of encoding for the e-mail message subject. Optional. Valid values: <ul style="list-style-type: none"> ◆ ASCII - plain English text ◆ Chinese, GB2312 ◆ Japanese, EUCV-JP ◆ Japanese, ISO 2022-JP ◆ Japanese, Shift JIS ◆ Korean, EUC-KR ◆ UTF-8 - 8 bit Unicode for English and Western European languages ◆ UTF-16 – 16 bit Unicode for Japanese and Chinese Note: Can also be configured in the GPM.
Request for a delivery receipt notification.	Delivery receipt requested. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes – (true) Requests delivery receipt notification. ◆ No – (false) Does not request delivery notification. Default. Note: Delivery notification only available for mail servers that support this option. Can also be configured in the GPM.
Request for a read notification.	Read notification requested. Optional. Valid values: <ul style="list-style-type: none"> ◆ Yes – (true) Requests read notification. ◆ No – (false) No read notification requested. Default. Note: Read notification only available for mail servers that support this option. Can also be configured in the GPM.
SMIME Encryption User Certificate	Valid user certificate. Optional.

Field	Description
SMIME Decryption User Certificate (System Store)	Not used in this release. Leave at default.
SSL	<p>The Secure Socket Layer (SSL) flag that turns on SSL socket negotiation. Optional.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ SSL_NONE – SSL socket negotiation is turned off. Default. ◆ SSL_MUST – SSL socket negotiation is turned on. <p>Note: Can also be configured in the GPM.</p>
Key Certificate Passphrase	Passphrase for the keystore if you are using SSL Client Authentication. Optional.
Cipher Strength	<p>The level of encryption applied to the message that flows through the socket connection. Optional.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ All – Default ◆ Strong ◆ Weak <p>Note: Can also be configured in the GPM.</p>
Key Certificate (System Store)	<p>Digital certificate that contains the private key and public certificate. The trading partner trusted certificate authority signs the certificate. Optional.</p> <p>Valid values: Any valid digital certificate in Application.</p> <p>Note: Can also be configured in the GPM.</p>
CA Certificates	<p>Digital certificate that contains the trusted certificate authority public certificate. Optional.</p> <p>Note: Can also be configured in the GPM.</p>

GPM Configuration

The following screen shows the graphical view of the GPM parameters for the SMTP Send adapter. The dimmed values were generated from the SMTP Send adapter configuration. All parameters are set on the Message To Service tab. No entries are required on the Message From Service tab.

Service Editor-SMTP Send Email		
Name	SMTP Send Email	
Config.	SMTP_SEND_ADAPTER	
<div style="display: flex; border-bottom: 1px solid black;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Message To Service</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">Message From Service</div> </div>		
Output Msg	Obtain Process Data first, then Messages	
Message Name	Xout	
Name	Value	Use XPATH?
b2b-raw-message		<input type="checkbox"/>
smtpAuthenticationOpt		<input type="checkbox"/>
smtpEncodingType		<input type="checkbox"/>
smtpHost	myhost	<input type="checkbox"/>
smtpnotifyDelivery		<input type="checkbox"/>
smtpnotifyRead		<input type="checkbox"/>
smtpPassword		<input type="checkbox"/>
smtpPort	25	<input type="checkbox"/>
smtpRetries	3	<input type="checkbox"/>
smtpRetryInterval	1	<input type="checkbox"/>
smtpSubject		<input type="checkbox"/>
smtpUser		<input type="checkbox"/>
xport-smtp-auth	true	<input type="checkbox"/>
xport-smtp-mailBCC	Recipient@company1.com	<input type="checkbox"/>
xport-smtp-mailCC	Recipient2@company2.com	<input type="checkbox"/>
xport-smtp-mailfrom	sender@company1.com	<input type="checkbox"/>
xport-smtp-mailhost	000.000.1.103	<input type="checkbox"/>
xport-smtp-mailport	25	<input type="checkbox"/>
xport-smtp-mailsubject	Document Notification	<input type="checkbox"/>
xport-smtp-mailsubjectencoding	ASCII	<input type="checkbox"/>
xport-smtp-mailto	receiver@company2.com	<input type="checkbox"/>
xport-smtp-notify-delivery	true	<input type="checkbox"/>
xport-smtp-notify-read	true	<input type="checkbox"/>
xport-smtp-retries	3	<input type="checkbox"/>
xport-smtp-retryinterval	1	<input type="checkbox"/>
xport-tp-cakeycert-id	server:105e37:f3be012345-52e9	<input type="checkbox"/>
xport-tp-cipher	STRONG	<input type="checkbox"/>
xport-tp-keycert-id	server:218e57:f3bd123456-733b	<input type="checkbox"/>
xport-tp-ssloption	SSL_MUST	<input type="checkbox"/>

All of the parameters beginning with *smtp* were either set or are defaults from the service configuration. To set the same values in the GPM or the business process, the following relationships exist:

SMTP Service Configuration (BPML value)	GPM Configuration (BPML value)
smtpAuthenticationOpt	xport-smtp-auth
smtpEncodingType	xport-smtp-mailsubjectencoding
smtpHost	xport-smtp-mailhost

SMTP Service Configuration (BPML value)	GPM Configuration (BPML value)
smtpnotifyDelivery	xport-smtp-notifyDelivery
smtpnotifyRead	xport-smtp-notifyRead
smtpPort	xport-smtp-Port
smtpSubject	xport-smtp-mailsubject

The following example shows the corresponding BPML parameters for the SMTP Send adapter as configured above.

```
<operation name="SMTP Send Email">
  <participant name="SMTP_SEND_ADAPTER" />
  <output message="Xout">
    <assign to="." from="*"></assign>
    <assign to="xport-smtp-auth">true</assign>
    <assign to="xport-smtp-mailBCC">Recipient@company1.com</assign>
    <assign to="xport-smtp-mailCC">Recipient2@company2.com</assign>
    <assign to="xport-smtp-mailfrom">sender@company1.com</assign>
    <assign to="xport-smtp-mailhost">000.000.1.103</assign>
    <assign to="xport-smtp-mailport">25</assign>
    <assign to="xport-smtp-mailsubject">Document Notification</assign>
    <assign to="xport-smtp-mailsubjectencoding">ASCII</assign>
    <assign to="xport-smtp-mailto">receiver@company2.com</assign>
    <assign to="xport-smtp-notify-delivery">true</assign>
    <assign to="xport-smtp-notify-read">true</assign>
    <assign to="xport-smtp-retries">3</assign>
    <assign to="xport-smtp-retryinterval">1</assign>
    <assign to="xport-tp-cakeycert-id">server:105e37:f3be012345:-52e9 </assign>
    <assign to="xport-tp-cipher">STRONG</assign>
    <assign to="xport-tp-keycert-id">server:218e57:f3bd123456:-733b </assign>
    <assign to="xport-tp-ssloption">SSL_MUST</assign>
  </output>
  <input message="Xin">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

The following table describes the fields used to configure the SMTP Send adapter in the GPM. This lists the fields that are only set in the GPM.

Note: The parameter names listed are also the corresponding BPML field names in the GPM.

Parameter Name	Description
Config	Name of the adapter configuration. Required.
xport-smtp-host	Trading partner host system IP address or host name. Required if not set in the Service Configuration. Valid values: The IP address or hostname of the mail server that will be sending the e-mail.
xport-smtp-port	Trading partner port number. Required if not set in the Service Configuration. Valid values: The port number used by the mail server sending the e-mail. The most commonly used port is 25.
xport-smtp-mailfrom	Originator e-mail address. Required when b2b-raw-message is set to false. Valid values: Valid e-mail address for the mail server used to send the e-mail. This address appears in the From line of the e-mail.
xport-smtp-mailto	Recipient e-mail address. Required when b2b-raw-message is set to false. Valid values: Valid e-mail address where the e-mail is being sent. This address appears in the To line of the e-mail.
xport-smtp-mailCC	CC list e-mail address. Optional. Valid values: The e-mail address to send a copy of the e-mail. This address appears in the CC line of the e-mail.
xport-smtp-mailBCC	BCC list e-mail address. Optional. Valid values: e-mail address to send a blind copy of the e-mail. This address appears in the BCC line of the e-mail.
xport-smtp-mailsubject	Subject of the message. Optional. This information appears in the SUBJECT line of the e-mail.
xport-smtp-auth	Valid values: <ul style="list-style-type: none">◆ true – The mail server requires SMTP authorization.◆ false – No SMTP authorization required. Default.
xport-tp-authfile	Added using the Advanced Editor. The authfile option is being deprecated but has been kept for backward compatibility. If SMTP authorization is required, set it in the service configuration.

b2b-raw-message	<p>Indicates if the primary document input to the adapter is a raw or non-raw message.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ true – Message is a raw message indicating that it is in internet-compliant format and that the document must contain the SMTP e-mail header instructions. ◆ false (default) – Input is not in internet-compliant format. The primary document becomes the plain-text body of the e-mail message.
xport-smtp-notify-delivery	<p>Delivery notification requested. Optional.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – (true) Requests delivery notification. ◆ No – (false) No delivery notification requested. Default. <p>Note: Delivery notification only available for mail servers that support this option.</p>
xport-smtp-notify-read	<p>Read notification requested. Optional.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ Yes – (true) Requests read notification. ◆ No – (false) No read notification requested. Default. <p>Note: Read notification only available for mail servers that support this option.</p>
xport-smtp-retries	<p>Number of times SMTP Send adapter tries to connect to the SMTP server if the connection is not successful. Optional.</p> <p>Valid value is an integer.</p>
xport-smtp-retryinterval	<p>Time delay between retrying the connection. Optional.</p> <p>Valid value is an integer.</p>
xport-smtp-mailsubjectencoding	<p>The type of encoding for the e-mail message subject.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ ASCII - plain English text ◆ Chinese, GB2312 ◆ Japanese, EUCV-JP ◆ Japanese, ISO 2022-JP ◆ Japanese, Shift JIS ◆ Korean, EUC-KR ◆ UTF-8 - 8 bit Unicode for English and Western European languages ◆ UTF-16 – 16 bit Unicode for Japanese and Chinese
xport-tp-ssloption	<p>SSL flag that turns on SSL socket negotiation.</p> <p>Valid values:</p> <ul style="list-style-type: none"> ◆ SSL_NONE – SSL socket negotiation is turned off. Default. ◆ SSL_MUST – SSL socket negotiation is turned on.

xport-tp-cipher	Level of encryption to be applied to the message document that flows through the socket connection. Valid values: <ul style="list-style-type: none"> ◆ ALL – Default. ◆ STRONG ◆ WEAK
xport-tp-cakeycert-id	Object ID (from the Trading Partner > Digital Certificate menu) that contains the trusted certificate authority public certificate.
xport-tp-keycert-id	Object ID (from the Trading Partner > Digital Certificate menu) that contains the PrivateKey and public certificate. The certificate is signed by the trading partner Trusted Certificate Authority.

SMTP Send Adapter Business Process Usage

This section contains additional examples using the SMTP Send adapter for sending e-mail. Examples are included both in the GPM and in BPML. The following business process examples illustrate various modes of operation for the SMTP Send adapter.

Non-raw Message Mode with SMTP authentication

The following example demonstrates using the SMTP Send adapter with a primary document that is not in raw message format. This document needs to be either a plain text document or one that is readable as a plain text document. The adapter creates an RFC822 compliant document to send with the primary document as the plain text body of the e-mail.

This example (using the GPM) illustrates a business process that performs an SMTP operation when the business process is started. The example illustrates the following:

The primary document is not in raw message format.

You are sending e-mail messages using mail host 00.000.00.00 on port 25.

The e-mail subject is set to “This is the subject.”

The originator is stest1@co.com.

The recipient is stest2@co.com.

The cc is set to stest3@co.com.

The bcc is set to stest4@co.com.

The connection to the mail host server is established over an SSL session, using client authentication.

- ◆ The certificates are retrieved from the system using Key cert-id = server:105e37:f3be012345:-52e9.

Service Editor-One

Name: One

Config: SMTP_SEND_ADAPTER

Message To Service: Message From Service

Output Msg: Obtain Process Data first, then Messages

Message Name: Xout

b2b-raw-message is set to false. The SMTP adapter constructs a message with the primary document as the plain-text body of the e-mail message.

Name	Value	Use XPATH?
b2b-raw-message	false	<input type="checkbox"/>
smtpAuthenticationOpt	No	<input type="checkbox"/>
smtpEncodingType		<input type="checkbox"/>
smtpHost	myhost	<input type="checkbox"/>
smtpnotifyDelivery		<input type="checkbox"/>
smtpnotifyRead		<input type="checkbox"/>
smtpPassword	*****	<input type="checkbox"/>
smtpPort	25	<input type="checkbox"/>
smtpRetries	3	<input type="checkbox"/>
smtpRetryInterval	1	<input type="checkbox"/>
smtpSubject		<input type="checkbox"/>
smtpUser	UserName	<input type="checkbox"/>
xport-smtp-auth	true	<input type="checkbox"/>
xport-smtp-mailBCC	stest4@co.com	<input type="checkbox"/>
xport-smtp-mailCC	stest3@co.com	<input type="checkbox"/>
xport-smtp-mailfrom	stest1@co.com	<input type="checkbox"/>
xport-smtp-mailhost	00.000.00.00	<input type="checkbox"/>
xport-smtp-mailport	25	<input type="checkbox"/>
xport-smtp-mailsubject	This is the subject	<input type="checkbox"/>
xport-smtp-mailsubjectencoding		<input type="checkbox"/>
xport-smtp-mailto	stest2@co.com	<input type="checkbox"/>
xport-smtp-notify-delivery		<input type="checkbox"/>
xport-smtp-notify-read		<input type="checkbox"/>
xport-smtp-retries		<input type="checkbox"/>
xport-smtp-retryinterval		<input type="checkbox"/>
xport-tp-cakeycert-id	server:105e37:f3be012345:-52e9	<input type="checkbox"/>
xport-tp-cipher		<input type="checkbox"/>
xport-tp-keycert-id	server:218e57:f3bd123456:-733b	<input type="checkbox"/>
xport-tp-ssloption	SSL_MUST	<input type="checkbox"/>

The following example shows the corresponding business process solution using BPML.

The following example shows a sample text message sent in the non-raw message format. It is the primary document and becomes the body of the message.

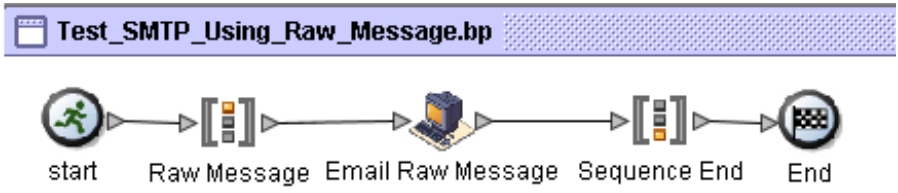
Raw Message Format

“Raw Message” refers to an Internet RFC822-compliant message as described in the parameter table. This message contains a message envelope and message content. The message envelope contains all the information needed to accomplish transmission and delivery of the message.

When the Raw Message Format option is selected, the primary document must be in Internet RFC822-compliant format. This message can be passed to or picked up by the business process in this format, or the message can be constructed dynamically by the business process. A common way to construct a simple dynamic message is through the use of XSLT (see XSLT Service page 713). This allows the message as well as the header (envelope) information to change as needed.

The following is an example of raw message format:

The following business process sends e-mail using mail host 00.000.00.00 on port 25. This configuration takes a raw document of the type described above or a multipart MIME document that is Internet RFC822-compliant (see *Multiple Documents (e-mail with multiple attachments)*).



Service Editor-Email Raw Message

Name: Email Raw Message

Config: SMTP_SEND_ADAPTER

Message To Service | **Message From Service**

Output Msg: Obtain Process Data first, then Messages

Message Name: Xout

Name	Value	Use XPATH?
b2b-raw-message	true	<input type="checkbox"/>
smtpAuthenticationOpt		<input type="checkbox"/>
smtpEncodingType		<input type="checkbox"/>
smtpHost	myhost	<input type="checkbox"/>
smtpnotifyDelivery		<input type="checkbox"/>
smtpnotifyRead		<input type="checkbox"/>
smtpPassword		<input type="checkbox"/>
smtpPort	00.000.00.00, port 25.	<input type="checkbox"/>
smtpRetries	This is used when the primary document is internet-compliant and no translation is needed.	<input type="checkbox"/>
smtpRetryInterval		<input type="checkbox"/>
smtpSubject		<input type="checkbox"/>
smtpUser		<input type="checkbox"/>
xport-smtp-auth		<input type="checkbox"/>
xport-smtp-mailBCC		<input type="checkbox"/>
xport-smtp-mailCC		<input type="checkbox"/>
xport-smtp-mailfrom		<input type="checkbox"/>
xport-smtp-mailhost	00.000.00.00	<input type="checkbox"/>
xport-smtp-mailport	25	<input type="checkbox"/>
xport-smtp-mailsubject		<input type="checkbox"/>
xport-smtp-mailsubjectencoding		<input type="checkbox"/>

Note: b2b-raw-message is set to true. The SMTP adapter takes the primary document and pushes it to mail server 00.000.00.00, port 25. This is used when the primary document is internet-compliant and no translation is needed.

The following example shows the corresponding business process solution using BPML.

Multiple Documents (e-mail with multiple attachments)

To send multiple documents in a single e-mail message, an RFC822-compliant MIME message must be constructed with all the documents included. This MIME message is sent by the SMTP Send adapter using raw message mode. Please refer to MIME Service document, Appendix A, for how to construct a RFC822 compliant MIME message.

In the event that a message of this type is used, the SMTP Send adapter is configured as in *Raw Message Format*.

B2B Mode

B2B mode is used when trading partner information is required as part of the application. There is a specialized application of the SMTP Send adapter that is available for use when working with B2B applications. See *B2B SMTP Client Adapter* for information on how to use this application.

For B2B applications and for transmitting EDI data, the required SMTP information can be specified in the trading profiles and accessed in a business process by using the B2B Send service. See *B2B Send System Service* for more information about the B2B Send service.

SOA Inbound Message Processing Service (Build 4303 or higher)

The SOA Inbound Message Processing service removes HTTP headers from the incoming SOAP request for both the Web service provider and consumer. It also removes attachments from MIME type messages in the case of consumers, and separates attachments from MIME messages in the case of providers. The following table provides an overview of the SOA Inbound Message Processing service:

System Name	SOA Inbound Message Processing Service
Graphical Process Modeler (GPM) categories	All Services
Description	The SOA Inbound Message Processing Service removes HTTP headers from the SOAP request for both the Web service provider and consumer. It also removes attachments from MIME type messages in the case of consumers, and separates attachments from MIME messages in the case of providers.
Business usage	Configured in GPM for an application Web service consumer process to remove HTTP headers from a SOAPs message after the response is received by the HTTP/HTTPS adapter. Used by the system business process WS_MessageHandler in Web service provider.
Usage example	On the consumer side of Web service, this service is used to remove HTTP headers from the SOAP response message. It also removes attachments from MIME type message after receiving the response from the application server through HTTP/HTTPS adapters. On the provider side, this service is used to remove the headers from SOAP request message and separate attachments from MIME messages before processing the request.
Preconfigured?	Yes. SOAInboundMsgProcessingService_Instance
Requires third party files?	None
Platform availability	All supported application platforms
Related services	Generally used after an HTTP/HTTPS adapter after receiving the request or response to the Web application server.
Application requirements	WS Licensing
Initiates business processes?	No
Invocation	This service is used in a GPM in case of Web service consumer (must be placed after an HTTP Adapter service). In the case of a Web service provider, it will be invoked internally by the business process used for provider to receive and process the SOAP request.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ Success – Service is able to understand the Soap message. ◆ Error – SOAP fault information.
Restrictions	There are no requirements on the usage of this service. The understanding of Web service provider and consumer will be offline and this service will come into the picture if the SOAP message is sent/received by the application.
Persistence level	System default
Testing considerations	<p>Debug information can be found by observing the execution flow of the business process in the application UI .The level of logging can be controlled through the log. properties file, just as for other log files.</p> <p>Also see Business Usage, Usage Example and Application Requirements in the previous sections.</p>

Implementing the SOA Inbound Message Processing Service

To implement the SOA Inbound Message Processing service, complete the following tasks:

1. Create a configuration of the SOA Inbound Message Processing service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the SOA Inbound Message Processing Service*.
2. Specify field settings for the service configuration in the application Admin Console and in the GPM as necessary.

Configuring the SOA Inbound Message Processing Service

You must specify field settings in the application, using the Admin Console, and in the GPM.

Creating or Setting Up a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the SOA Inbound Message Processing service, or to set up the configuration provided with the application. Some fields are available in both the Admin Console and in the GPM. For the fields that are available in both, the GPM field name is shown in parentheses in the table below.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list <p>Note: See <i>Managing Services and Adapters</i>.</p>

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.

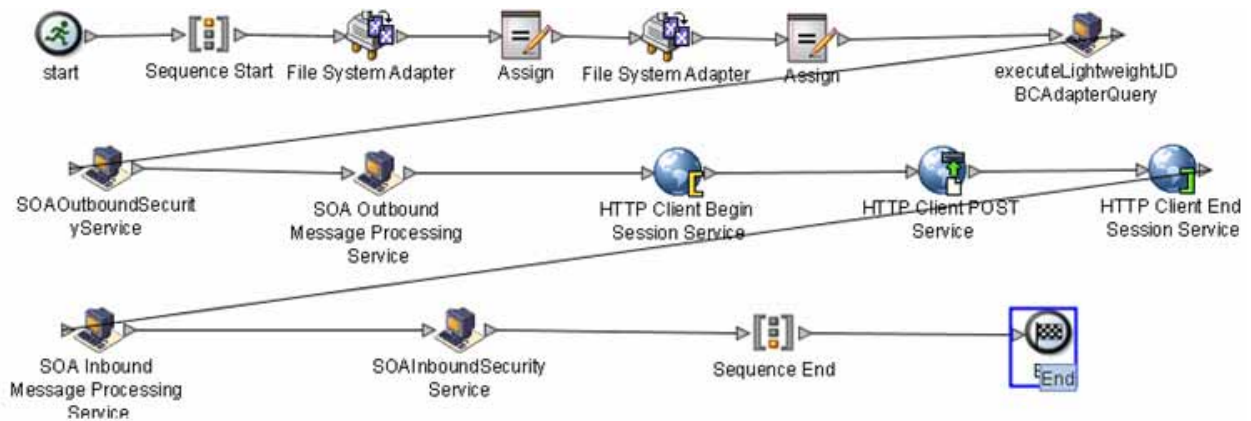
Business Process Example

The following example illustrates how the SOA Inbound Message Processing service can be used in a business process:

Example Business Process as Web Service Provider

The following business process uses a dynamic service created in an application for sending a SOAP message through Http to an endpoint configured in the application itself (where the application is the consumer as well as provider). The SOA Inbound Message Processing Service is used after the WS Config Info service.

Example Business Process as WebService Consumer



Parameters Passed From Service to Business Process

The following table contains the parameters passed from the SOA Inbound Message Processing service to the business process:

Parameter	Description
AttachmentCID	The unique ID (the MIME content ID) of the attachment, if any, that is passed with the SOAP Message.
Primary Document	Primary Document
SOAP Document Part	SOAP Document part of the SOAP message

SOA Inbound Security Service (Build 4303 or higher)

This service is used by the Web services provider as well as Web services consumer to process the security part of an incoming SOAP message. This can be a request as well as response for provider and consumer, respectively. The processing may involve decryption, signature verification, and security token validation, although it is not necessary that all should be present.

The following table provides an overview of the SOA Inbound Security service:

System Name	SOA Inbound Security Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service is used by the Web services provider as well as Web services consumer to process the Security part of incoming SOAP message. This can be a request as well as response for provider and consumer, respectively. The processing involves decryption, signature verification and security token validation, although it is not necessary that all should be present.
Business usage	In case the SOAP message (either request or response) coming to the application is a secure message, this service is used to process the message to make it compatible with succeeding nodes/services for further processing.
Usage example	<p>On a consumer side of Web services, the response received is a secure SOAP message. In that case, this service is used to understand that secure response which includes decryption, signature validation, and security token validation.</p> <p>On the provider side, the request received is a secure SOAP message. In that case, this service is used to understand that secure request which includes decryption, signature validation, and security token validation.</p>
Preconfigured?	Yes. SOAInboundSecurityService_Instance.
Requires third party files?	No third party sources are required. All the required jars (wss4j and xmlSecurity 1.3) are shipped with the application install
Platform availability	All supported application platforms
Related services	This service is used by the SOA Inbound Message Processing Service to process the security header part of an Incoming SOAP Message.
Application requirements	<p>Private Certificate for decryption.</p> <p>Public Certificate for signature verification and security token.</p> <p>The document passed to this must be a proper SOAP document.</p>
Initiates business processes?	No
Invocation	<p>This service is invoked from a business process in case of Web service consumer (must be put after SOA Inbound Message Processing Service).</p> <p>In case of a Web service provider, it will be invoked internally by the business process used for provider to receive and process the SOAP request.</p>

Business process context considerations	For consumer and provider, the SOA Inbound Security Service allows the user to specify the decryption certificate (private certificate), Signature Verification certificate (public certificate) and security token for processing the incoming secure SOAP message.
Returned status values	<ul style="list-style-type: none"> ◆ Success – Service is able to understand Soap message. ◆ Error – Some error occurs while processing the security header. This can be due to incorrect certificate configurations.
Restrictions	There are no restrictions on the usage of this service. The understanding about the security between a Web service provider and consumer is offline and this service is used if the SOAP message is a secure SOAP message.
Persistence level	System default
Testing considerations	Debug information can be found either in by observing the execution flow of the business process in the application user interface or using the wssec.log. The level of logging is controlled through the log. properties file, just as for other log files.

Implementing the SOA Inbound Security Service

To implement the SOA Inbound Security service, complete the following tasks:

1. Create a configuration of the SOA Inbound Security service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the SOA Inbound Security Service*.
2. Specify field settings for the service configuration in the application Admin Console and in the GPM as necessary.

Configuring the SOA Inbound Security Service

You must specify field settings in the application, using the Admin Console, and in the GPM.

Creating or Setting Up a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the SOA Inbound Security service, or to set up the configuration provided with the application. Some fields are available in both the Admin Console and in the GPM. For the fields that are available in both, the GPM field name is shown in parentheses.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list <p>Note: See <i>Managing Services and Adapters</i>.</p>

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.

Configuration Parameters (Consumer)

Following are the parameters that can be configured in SOA Outbound Message Processing Service for adding various security headers to the SOAP message. For detailed information about the possible values, refer to the WS-Security specification:

Parameter	Description
Name	Name that the service will have in the application
Description	Description of the service
Select a Group	Radio button that allows selection or creation of the group that this service will be part of

Parameters Provided in Request Security Settings (Provider)

Parameter	Description
Decryption Certificate	Name of private certificate used to decrypt the SOAP message
UserName Tokens	Name of security token checked in the application used for validating the token in the incoming soap message
Verification Certificate	Name of public certificate which is used to verify the signed SOAP request

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the SOA Inbound Security service:

Parameter	Description
DECRYPTION_KEY	Name of private certificate used to decrypt the soap message (in BPML, the certificate ID).
SECURITY_TOKEN_NAME	Name of security token checked into the application. It is used for validating the token in the incoming SOAP message.
VERIFICATION_KEY	Name of public certificate used to verify the signed SOAP request (in BPML, the certificate ID).

Parameters Passed From Service to Business Process

The output will be a SOAP message with the security header processed and (if the original SOAP message was encrypted) the body decrypted.

Business Process Example

Application consumer:

SOA Outbound Message Processing Service (Build 4303 or higher)

The SOA Outbound Message Processing service adds required HTTP headers to the SOAP message, which can then be processed by HTTP/HTTPS adapters before sending request or response by Web service consumer and provider, respectively. It also converts the attachments to MIME type. The following table provides an overview of the SOA Outbound Message Processing service:

System Name	SOA Outbound Message Processing Service
Graphical Process Modeler (GPM) categories	All Services
Description	The SOA Outbound Message Processing Service adds required HTTP headers to the SOAP message, which can then be processed by HTTP/HTTPS adapters before sending request or response by Web service consumer and provider, respectively. It also converts the attachments to MIME type.
Business usage	Configured in GPM for an application Web Service consumer process to add required headers to soap message before it is being posted by the HTTP/HTTPS adapter. Used by the system business process WS_RequestHandler in Web service provider.
Usage example	On the consumer side of the Web service, this service is used to add headers to the SOAP request message and convert attachments to MIME type before sending request to the application server through HTTP/HTTPS adapters. On the provider side, this service is used to add headers to the SOAP response message and convert attachments to MIME type before sending a response to the application server through HTTP/HTTPS adapters.
Preconfigured?	Yes. SOAOutboundMsgProcessingService_Instance.
Requires third party files?	None
Platform availability	All supported application platforms
Related services	Generally used before an HTTP/HTTPS adapter before sending the request or response to the Web application server.
Application requirements	WS Licensing
Initiates business processes?	No
Invocation	This service is used in a GPM in case of Web service consumer (must be put before HTTP adapter service) In case of Web service provider, it will be invoked internally by the business process used for provider to receive and process the SOAP request.
Business process context considerations	None

Returned status values	<ul style="list-style-type: none"> ◆ Success – Service is able to understand the SOAP message. ◆ Error – SOAP Fault information.
Restrictions	There are no requirements on the usage of this service. The understanding of Web service provider and consumer will be offline and this service will come into the picture if the SOAP message is sent/received by the application.
Persistence level	System default
Testing considerations	<p>Debug information can be found by observing the execution flow of the business process in the application user interface. The level of logging is controlled through the log.properties file, just as for other log files.</p> <p>Also see Business usage, Usage example and Application requirements in the previous sections.</p>

Implementing the SOA Outbound Message Processing Service

To implement the SOA Outbound Message Processing service, complete the following tasks:

1. Create a configuration of the SOA Outbound Message Processing service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the SOA Outbound Message Processing Service*.
2. Specify field settings for the service configuration in the application Admin Console and in the GPM as necessary.

Configuring the SOA Outbound Message Processing Service

You must specify field settings in the application, using the Admin Console, and in the GPM.

Creating or Setting Up a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the SOA Outbound Message Processing service, or to set up the configuration provided with the application. Some fields are available in both the Admin Console and in the GPM. For the fields that are available in both, the GPM field name is shown in parentheses in the table below.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list <p>Note: See <i>Managing Services and Adapters</i>.</p>

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.

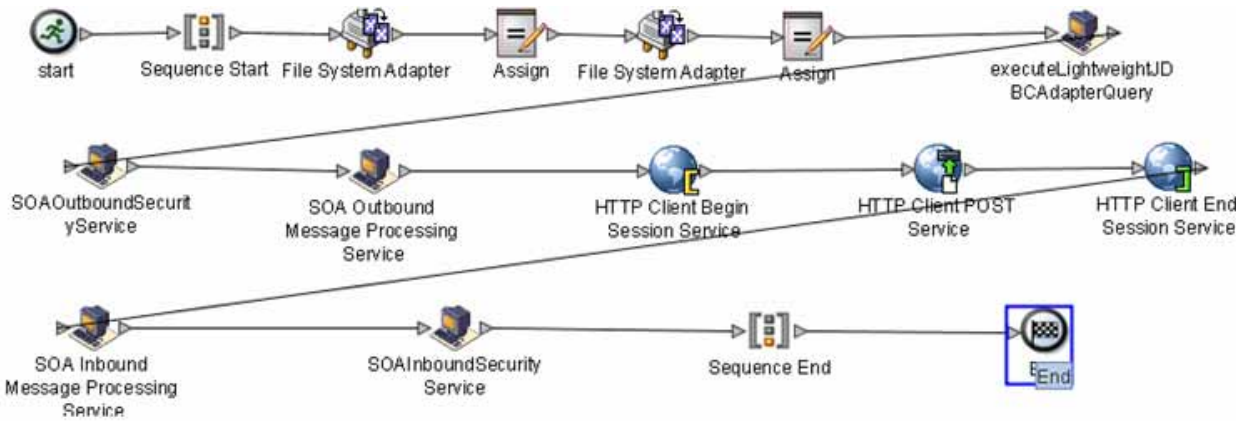
Business Process Example

The following example illustrates how the SOA Outbound Message Processing service can be used in a business process:

Example Business Process as WebService Provider

The following business process uses a dynamic service created in the application for sending a SOAP message through HTTP to an endpoint configured in the application itself (where the application is the consumer as well as provider). The SOA Outbound Message Processing Service is used before the HTTP service.

Example Business Process as Web Service Consumer



Parameters Passed From Service to Business Process

The following table contains the parameters passed from the SOA Outbound Message Processing service to the business process:

Parameter	Description
Attachment ID	Unique ID (MIME content ID) of the attachment, if any, that is passed with the SOAP message
Primary Document	Primary Document
SOAP Document Part	SOAP Document part of the SOAP message

SOA Outbound Security Service (Build 4303 or higher)

This service adds security headers to a SOAP envelope as configured in the Graphical Process Modeler (GPM). The following table provides an overview of the SOA Outbound Security service:

System Name	SOA Outbound Security Service
Graphical Process Modeler (GPM) categories	All Services
Description	This service adds security headers to a SOAP envelope as configured in GPM.
Business usage	The primary business usage of this service will be to enable Web services Security in any outbound SOAP message. A user will be able to add WS-Security related headers to any SOAP message. WS-Encryption, Signature etc. provided by this service will enable the user to make use of message-level security features for SOAP messages.
Usage example	This service would be used mainly in a business process which has a SOAP envelope as a primary document. The SOAP envelope will have additional SOAP headers (WS-Security headers) and possibly modified SOAP body (encryption) after being processed by this service.
Preconfigured?	Yes. SOAOutboundSecurityService_Instance.
Requires third party files?	The WS-security features depend upon wssj4 1.5.1 and xmlSecurity 1.3 which are open source modules. These are shipped with the application so that the user doesn't have to add this explicitly.
Platform availability	All supported application platforms
Related services	Generally used before the SOA Outbound Message Processing service, which adds required Http Headers to the SOAP message, which can then be processed by http adapters.
Application requirements	The certificates used for encrypting and signing have to be either obtained from the other party or generated in your application. The obtained certificates need to be checked into your application as trusted certificates before configuring this service in GPM. Also, it is assumed that the Primary Document passed to this service is a properly formatted SOAP envelope.
Initiates business processes?	No
Invocation	Can be invoked by putting in any business process. Configure the parameters in GPM and pass a proper SOAP envelope to this service.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – Security headers were added to the envelope.◆ Error – Some parameters were wrongly configured or the SOAP envelope passed was malformed.

Restrictions	All required parameters have to be entered in GPM (depending upon what headers are needed.)
Persistence level	System default
Testing considerations	See Business usage, usage Example and Application Requirements above.

Implementing the SOA Outbound Security Service

To implement the SOA Outbound Security service, complete the following tasks:

1. Create a configuration of the SOA Outbound Security service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the SOA Outbound Security Service*.
2. Specify field settings for the service configuration in the application Admin Console and in the GPM as necessary.

Configuring the SOA Outbound Security Service

You must specify field settings in the application, using the Admin Console, and in the GPM.

Creating or Setting Up a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the SOA Outbound Security service, or to set up the configuration provided with the application. Some fields are available in both the Admin Console and in the GPM. For the fields that are available in both, the GPM field name is shown in parentheses in the table below.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a service group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) ◆ Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list <p>Note: See <i>Managing Services and Adapters</i>.</p>

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.

GPM Parameters

Following are the parameters that can be configured in SOA Outbound Security Service for adding various Security headers to the SOAP message. For detailed information about values for these parameters, please refer to the WS-Security specification.

Parameter	Description
SIGNING_CERTIFICATE	Private key used for signing SOAP message. This will be a System Certificate ID. A list will be displayed to the user in GPM.
SIGNING_ALGO	Signature algorithm to be used for signing the SOAP message.
SIGNING_KEY_IDENTIFIER_TYPE	Identifier type to refer the signing key as, in the signature info in the wsse:Security header.
CANONICALIZATION_ALGO	Canonicalization (formatting) to be applied to the SOAP message before signing it so that signature value is interoperable.
ENCRYPTION_CERTIFICATE	Public key to be used for encryption. This will be a trusted certificate checked into your application. A list will be displayed to the user in GPM.
KEY_ENCODING_ALGO	The "Asymmetric" key algorithm used to encode the keys used in symmetric key encryption (which is the actual algorithm used for encryption of the SOAP message). This algorithm is used to only encode the keys and not the SOAP message.
SYMMETRIC_KEY_ALGO	Symmetric Key Algorithm used to encrypt the SOAP message.
ENCRYPT_KEY_IDENTIFIER_TYPE	Identifier type for the encryption key in the wsse:Security header.
SIGNATURE_ENCRYPTION_ORDER	Specifies whether to Encrypt First or Sign First.
INSERT_USER_NAME_TOKEN	Specifies whether a user name token should be added to the security header.
USER_NAME_TOKEN_NAME	User name token to be used for the security header. For this dropdown to be populated, at least one Security token must be configured for Web services (Deployment > Webservices > SecurityToken).
INSERT_TIME_STAMPS	Specified whether a timestamp security header needs to be inserted.
TIME_TO_LIVE	Time interval value used in the timestamp header specified above.
SECURITY_HEADER_ACTOR	Value of the Actor attribute that can be set in the wsse:Security header element.

Parameter	Description
INSERT_MUSTUNDERSTAND	Specifies whether to add the mustUnderstand=1 attribute to the wsse:security header. If this is set, processing of the security header will be made mandatory for the receiving party.

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the SOA Outbound Security service to the business process:

Parameter	Description
SOAPEnvNSPrefix	The prefix used for SOAP envelope element
SOAPEnvNSURI	The Namespace URI for the above prefix

Business Process Example

The following business process uses a dynamic service created in the application for sending a SOAP message through HTTP to an endpoint also configured in the application (that is, the application is the consumer as well as provider). The parameters in the business process in SOA Outbound Security Service can be configured using GPM.

```

<assign to="LightweightJDBCAdapterQuery"
from="//PrimaryDocument/@SCIOBJECTID"></assign>
<operation name="File System Adapter">
  <participant name="test_fsa"/>
  <output message="FileSystemInputMessage">
    <assign to="Action">FS_COLLECT</assign>
    <assign to="attachFile">mesaAuth.txt</assign>
    <assign to="deleteAfterCollect">false</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>

```

```

    </input>
  </operation>

  <assign to="mesaAuth" from="//PrimaryDocument/@SCIOBJECTID"></assign>
  <operation name="executeLightweightJDBCAdapterQuery">
    <participant name="DS_MYTEST5_HTTPS_PORT1_OPE1_Instance"/>
    <output message="DS_MYTEST5_HTTPS_PORT1_OPE1InputMessage">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="SOAOutboundSecurityService">
    <participant name="outbsec_11sec"/>
    <output message="SOAOutboundSecurityServiceInputMessage">
      <assign
to="CANONICALIZATION_ALGO">http://www.w3.org/2001/10/xml-exc-c14n#</assign>
      <assign to="ENCRYPT_KEY_IDENTIFIER_TYPE">2</assign>
      <assign
to="ENCRYPTION_CERTIFICATE">blrgislin29:10.11.23.53:114f8acc4f3:13</assign>
      <assign to="INSERT_MUSTUNDERSTAND">0</assign>
      <assign to="INSERT_TIME_STAMPS">1</assign>
      <assign to="INSERT_USER_NAME_TOKEN">1</assign>
      <assign
to="KEY_ENCODING_ALGO">http://www.w3.org/2001/04/xmlenc#rsa-1_5</assign>
      <assign to="SECURITY_HEADER_ACTOR">actor1</assign>
      <assign to="SIGNATURE_ENCRYPTION_ORDER">0</assign>
      <assign to="SIGNING_ALGO">http://www.w3.org/2000/09/xmldsig#rsa-sha1</assign>
      <assign
to="SIGNING_CERTIFICATE">blrgislin29:10.11.23.53:114cef6bda9:1363</assign>
      <assign to="SIGNING_KEY_IDENTIFIER_TYPE">1</assign>
      <assign
to="SYMMETRIC_KEY_ALGO">http://www.w3.org/2001/04/xmlenc#aes128-cbc</assign>
      <assign to="TIME_TO_LIVE">0</assign>
      <assign to="USER_NAME_TOKEN_NAME">admin</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
  <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="SOA Outbound Message Processing Service">
    <participant name="SOAOutboundMsgProcessingService_Instance"/>
    <output message="SOAOutboundMsgProcessingTypeInputMessage">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="HTTP Client Begin Session Service">
    <participant name="HTTPClientBeginSession"/>

```

```

<output message="HTTPClientBeginSessionServiceTypeInputMessage">
  <assign to="HTTPClientAdapter">HttpClientAdapter_DynamicService</assign>
  <assign to="RemoteHost">10.11.23.53</assign>
  <assign to="RemotePort">9541</assign>
  <assign to="." from="*"></assign>
</output>
<input message="inmsg">
  <assign to="." from="*"></assign>
</input>
</operation>

<operation name="HTTP Client POST Service">
  <participant name="HTTPClientPost"/>
  <output message="HTTPClientPostServiceTypeInputMessage">
    <assign to="RawRequest">>true</assign>
    <assign to="RawResponse">>true</assign>
    <assign to="URI">/soap-new?service=Webservice_Test</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="HTTP Client End Session Service">
  <participant name="HTTPClientEndSession"/>
  <output message="HTTPClientEndSessionServiceTypeInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="SOA Inbound Message Processing Service">
  <participant name="SOAInboundMsgProcessingService_Instance"/>
  <output message="SOAInboundMsgProcessingServiceInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="SOAInboundSecurityService">
  <participant name="inbsec_11sec"/>
  <output message="SOAInboundSecurityServiceInputMessage">
    <assign to="DECRYPTION_KEY">blrgislin29:10.11.23.53:114cef6bda9:770</assign>
    <assign to="SECURITY_TOKEN_NAME">admin</assign>
    <assign to="VERIFICATION_KEY">blrgislin29:10.11.23.53:114f8acc4f3:8</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

```

```
</sequence>  
</process>
```

SOAP Inbound Service

The following table provides an overview of the Simple Object Access Protocol (SOAP) Inbound service:

System Name	SOAPInbound
Graphical Process Modeler (GPM) categories)	All Services, Internet B2B > SOAP
Description	<p>SOAP is an XML-based protocol that allows access to services, objects, and servers in a platform-independent manner.</p> <p>The SOAP Inbound Service is responsible for handling SOAP requests and responses. For SOAP requests, the service uses the URI to which the document was posted to set the business process that should be invoked to handle the request. The service also adds any SOAP headers targeted at it to process data, removes the SOAP enveloping from the request, and makes the SOAP payload the primary document. For SOAP responses, the services checks to see if it is expecting a SOAP response, and if it is, extracts the payload and makes it the primary document. It also checks to see if a SOAP Fault is present. If one is present, the service sets the workflow status to error.</p>
Business usage	Processes inbound SOAP messages.
Usage example	<ul style="list-style-type: none">◆ To receive SOAP messages, see <i>HandleSOAPRequest Business Process</i>.◆ To send SOAP messages, see <i>SendSOAPRequest Business Process</i>.
Preconfigured?	No. A configuration of the service, SOAPInbound, is installed with the application, but you must edit the service and define URIs for the service to use.
Requires third party files?	No
Platform availability	All supported application platforms
Related services	SOAP Outbound service, HTTP Server adapter, HTTP Client adapter Note: The HTTP Server adapter and HTTP Client adapter replace the B2B HTTP Server adapter and HTTP Send service, respectively, which have entered the retirement process. For more information on the retirement process, see <i>Retiring and Removed Services and Adapters</i> .
Application requirements	None
Initiates business processes?	Yes, you can specify a business process to start when a SOAP message is received.
Invocation	To invoke this service in the predefined business process called HandleSOAPRequest, configure a HTTP Server adapter with a URI mapped to the HandleSOAPRequest business process, specifying raw messages.
Business process context considerations	None
Returned status values	Success – Request or Response was handled without errors. Error – Errors were encountered somewhere in the SOAP document, and the advanced status will be set to SOAP-FAULT.

Restrictions	None
Testing considerations	Use the predefined business processes HandleSOAPInbound and SendSOAPRequest to test the service configuration.
Notes	<p>Output parameters are:</p> <ul style="list-style-type: none"> ◆ SOAP_URI – The URI to which the message was posted (bootstrap mode only) ◆ NextSoapURL – Where the message should be sent next (if intermediate mode and bootstrap mode) ◆ ReceivedSOAPHeaders – The node containing the SOAP headers targeted at the service ◆ SOAPfaultcode – Could be present if a SOAP fault was encountered ◆ SOAPfaultstring – Could be present if a SOAP fault was encountered ◆ SOAP_INTERMEDIATE_NODE – “true” if the URI configuration says to act as an intermediate node. ◆ WFD_NAME – The definition name of the business process to invoke to actually handle the SOAP payload. ◆ SOAPActor – The SOAPActor specified in the configuration for the URI.

Implementing the SOAP Inbound Service

1. Before you configure the SOAP Inbound service, configure a corresponding HTTP Server adapter. When you configure the HTTP Server adapter:
 - ◆ Use the same Uniform Resource Indicator (URI) that you will use in the SOAP Inbound service.
 - ◆ Specify Raw Messages.
 - ◆ For the business process, specify HandleSOAPRequest.

Note: The HTTP Server adapter and HTTP Client adapter replace the B2B HTTP Server adapter and HTTP Send service, respectively, which have entered the retirement process. For more information on the retirement process, see *Retiring and Removed Services and Adapters*.

2. Create a SOAP Inbound service configuration, or edit the SOAP Inbound service configuration supplied with the application.

Note: If you create a new configuration, you must also create new copies of the predefined business processes, HandleSOAPRequest.bp and SendSOAPRequest.bp, and edit them to use the new service configuration.
3. Configure the SOAP Inbound service. See *Configuring the SOAP Inbound Service*.
4. Configure one or more of the predefined SOAP business processes to enable the application to receive or send SOAP messages.
 - ◆ To receive SOAP messages, see *HandleSOAPRequest Business Process*.
 - ◆ To send SOAP messages, see *SendSOAPRequest Business Process*.

Configuring the SOAP Inbound Service

To configure the SOAP Inbound service, you must specify settings for the following fields in the application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
URI	Universal resource identifier (URI) where others can post SOAP messages to the application. For each URI in the SOAP Inbound service, you must have corresponding URIs defined in an HTTP Server adapter.
Is this an intermediate node? Y or N	Whether the application is the ultimate receiver. Valid values are Yes and No.
Business Process	Business process to run when the application receives a SOAP message. Note: Do not select HandleSOAPRequest because the SOAP Inbound service runs as a result of that business process.
Actor	Unique name that identifies the application as a unique node. This is usually a URL.
Payload Root Tag	Valid XML tag that contains multiple payloads as if they were one payload.
Forward To	URL of the next node, if the application is an intermediate node.
Verify signature with	System certificate to verify the signature.
Expect signed messages? Y or N	Whether you expect to receive SOAP messages that are digitally signed. If you receive a message with a certificate expecting a signature and you do not receive a signature, the application generates a SOAP fault message. Also, if you receive a SOAP message with a signature, whether or not it is expected, and the signature does not verify, the application generates a SOAP fault.

The following table describes the fields used to configure a header for the SOAP Inbound service in the application:

Field	Description
Header	Header that the application can process. Add a new header for each header that the application accepts.
Namespace	Namespace that is related to the header.

HandleSOAPRequest Business Process

The HandleSOAPRequest business process enables the application to receive SOAP messages.

When HandleSOAPRequest receives a SOAP message, it validates and reads the message to determine the receiver of the message. If the application is the receiver of the message, HandleSOAPRequest completes the following process:

1. Verifies the digital signature.
2. Removes the SOAP envelope and puts any attachments and SOAP header blocks into process data. The application puts inbound SOAP message attachments in process data under the node SOAPRequestAttachments. The attachments follow this naming convention: SOAPAttachment1, SOAPAttachment2, and so on. The application sets the content ID attribute to the value specified in the SOAP message. See *Handling Multiple Incoming Attachments*.
3. Starts subprocesses associated with the URI to which the inbound message was sent (configured in the SOAP Inbound service).
4. Applies a SOAP envelope, if necessary, and HTTP headers to the document.
5. Returns a response to the requester.

If the application is not the receiver of the message, HandleSOAPRequest completes the following process:

1. Verifies the digital signature.
2. Removes SOAP header blocks that are specific to the application and puts them into process data.
3. Forwards a document to the next receiver.
4. Receives a response from the receiver.
5. Returns a response to the requester.

Before you use the HandleSOAPRequest business process, you must complete the following tasks:

- ◆ Configure the SOAP Inbound service.
- ◆ Configure a related HTTP Server adapter so that it does the following:
 - ◆ Uses raw messages
 - ◆ Specifies the same URI as the SOAP Inbound service
 - ◆ Specifies the HandleSOAPRequest business process

- ◆ Create the business process that processes the SOAP message where the application is the receiver of the message. Depending on your needs, you may need to assign values to several parameters in this business process before using the HandleSOAPRequest business process.

The following table identifies parameters for various SOAP functions and the values you may need to specify:

Required SOAP Function	SOAP Parameter	Parameter Values
SOAP enveloping on primary document before posting response	ADD_SOAP_ENVELOPING	true – Add SOAP enveloping (default).
Digitally sign a SOAP response	SIGN_WITH_KEY	Exact name of the digital certificate
Envelope namespace prefix for SOAP message	SOAPEnvNSPrefix	Envelope namespace prefix. ADD_SOAP_ENVELOPING must also be true. The default is SOAP_ENV.
Distinguish whether response is a SOAP message	RESPOND_NON_SOAP	true – Response is not a SOAP message. false – Response is a SOAP message.
Envelope namespace for SOAP message	SOAPEnvNSURI	Envelope namespace. The default is: http://www.w3.org/2001/06/soap-envelope

Business Process Example – HandleSOAPRequest

The following is the predefined HandleSOAPRequest business process, as delivered with the application:

```
<process name="HandleSOAPRequest">
  <rule name="acting_as_intermediate">
    <condition>SOAP_INTERMEDIATE_NODE = "true"</condition>
  </rule>
  <rule name="acting_as_endpoint">
    <condition>SOAP_INTERMEDIATE_NODE = "false"</condition>
  </rule>

  <sequence>
    <sequence>
      <operation name="SoapIn">
        <participant name="SOAPInbound"/>
        <output message="output">
          <assign to="." from="*"></assign>
        </output>
        <input message="input">
          <assign to="." from="*"></assign>
        </input>
      </operation>

      <onFault code="SOAP-FAULT">
        <assign to="InvokeSubBP">false</assign>
      </onFault>

      <choice>
        <select>
```

```

    <case ref="acting_as_intermediate" activity="act_as_intermediate" />
</select>

<sequence name="act_as_intermediate">
  <assign to="xport-http-url" from="NextSoapURL/text()"></assign>
  <assign to="b2b-message-mode">send</assign>

  <sequence>
    <operation name="HTTPSend">
      <participant name="SOAP_HTTP_Send" />
      <output message="Xout">
        <assign to="b2b-raw-response">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <onFault>
      <sequence>
        <assign to="SOAPfaultcode">Server</assign>
        <assign to="SOAPfaultstring">There was an error processing the SOAP
request.</assign>
        <assign to="SOAPdetail">An error occurred trying to send SOAP request
to next node.</assign>
      </sequence>
    </onFault>
  </sequence>

  <operation name="SoapIn">
    <participant name="SOAPInbound" />
    <output message="output">
      <assign to="." from="*"></assign>
      <assign to="bootstrap">>false</assign>
    </output>
    <input message="input">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <assign to="SOAPOutboundAttachments" from="SOAPResponseAttachments/*" />
</sequence>
</choice>

<choice>
  <select>
    <case ref="acting_as_endpoint" activity="act_as_endpoint" />
  </select>

  <sequence name="act_as_endpoint">
    <operation>
      <participant name="InvokeBusinessProcessService" />
      <output message="Xout">
        <assign to="." from="*"></assign>

```

```

        <assign to="INVOKE_MODE">SYNC</assign>
    </output>
    <input message="Xin" >
        <assign to="." from="*"></assign>
    </input>
</operation>

<onFault>
    <sequence>
        <assign to="SOAPfaultcode">Server</assign>
        <assign to="SOAPfaultstring">There was an error processing the SOAP
request.</assign>
        <assign to="SOAPdetail">An error occurred while processing the SOAP
request.</assign>
    </sequence>
</onFault>
</sequence>
</choice>

</sequence>

<operation name="SoapOut">
    <participant name="SOAPOutbound"/>
    <output message="output">
        <assign to="." from="*"></assign>
        <assign to="SOAP_MODE">respond</assign>
    </output>
    <input message="input">
        <assign to="." from="*"></assign>
    </input>
</operation>

<assign to="doc-has-headers">true</assign>

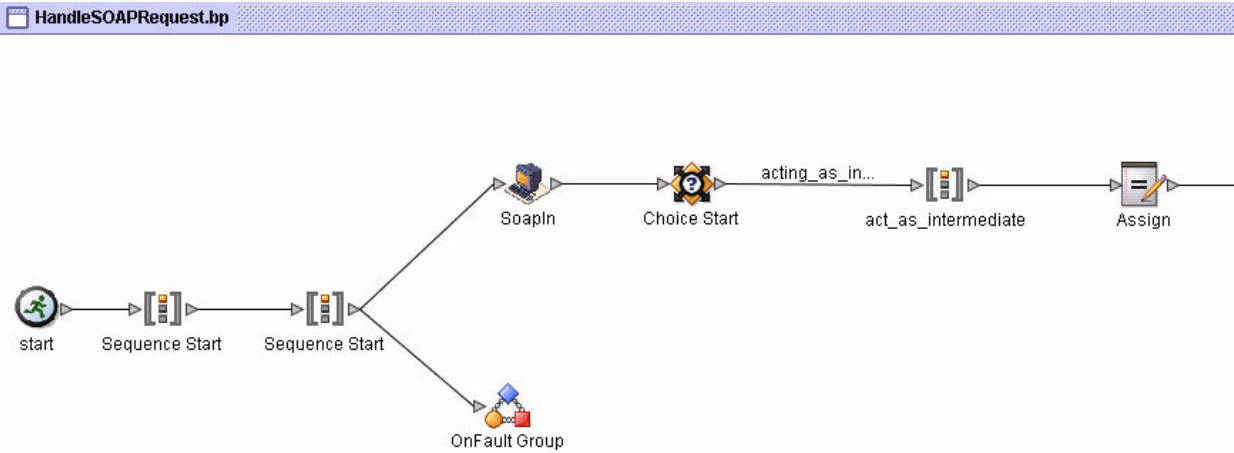
<operation name="HttpResponse">
    <participant name="HttpRespond"/>
    <output message="Xout">
        <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
        <assign to="." from="*"></assign>
    </input>
</operation>

</sequence>
</process>

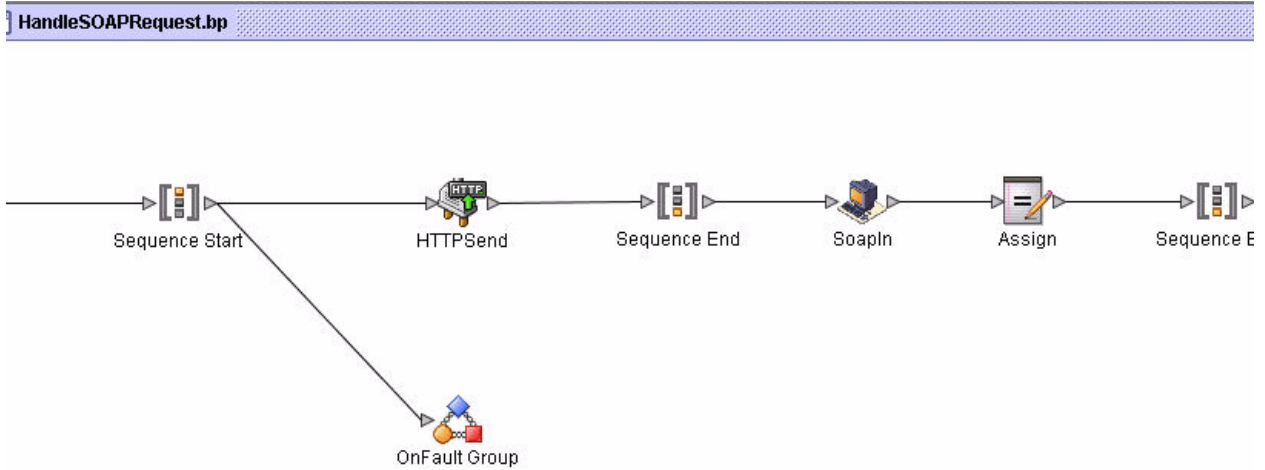
```

The following illustrations show the business process as it appears in the GPM:

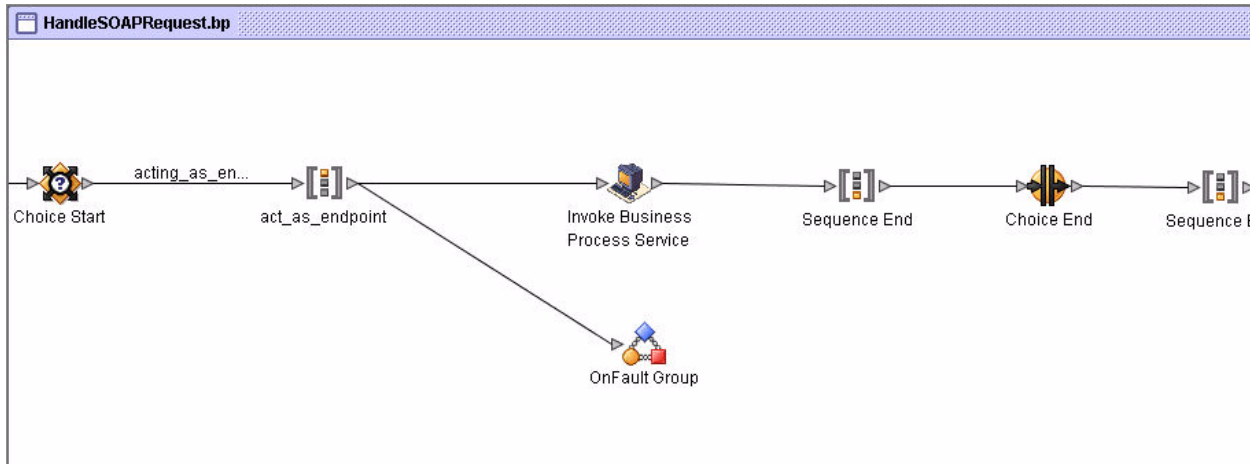
HandleSOAPRequest Screen 1 of 4



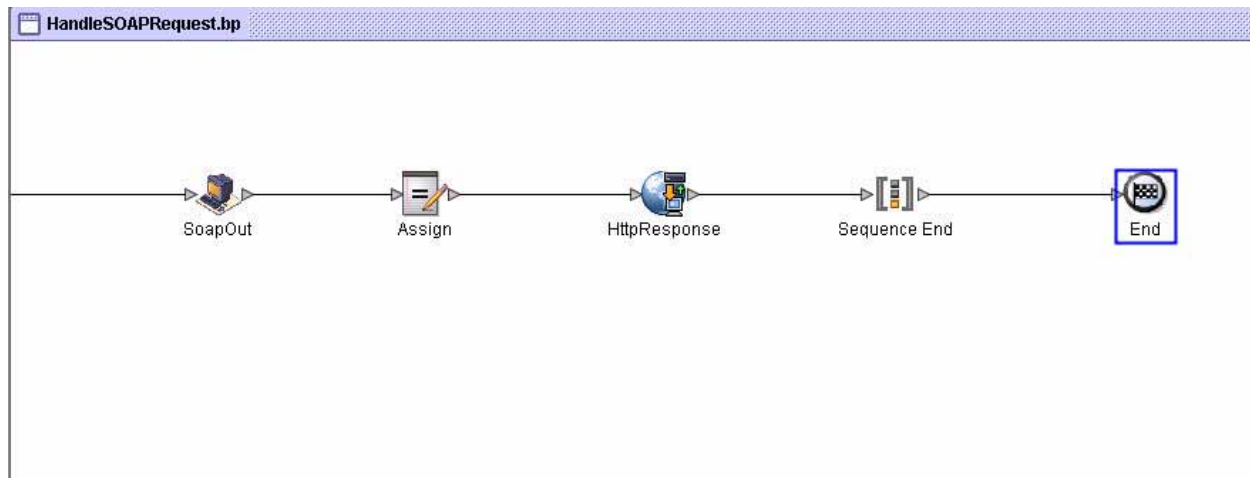
HandleSOAPRequest Screen 2 of 4



HandleSOAPRequest Screen 3 of 4



HandleSOAPRequest Screen 4 of 4



SendSOAPRequest Business Process

The SendSOAPRequest business process enables the application to send SOAP messages.

When the application sends a SOAP message, SendSOAPRequest completes the following process:

1. Applies a SOAP envelope to the data, if requested.
2. Adds SOAP headers to an outbound request if the process data contains SOAP headers. The business process takes all subordinate nodes under the SOAPHeaders node in process data and adds them as SOAP header blocks.
3. Adds SOAP attachments to an outbound request if the process data contains SOAP outbound attachments. The business process takes all subordinate nodes under the SOAPAttachments node,

which are named SOAPAttachment1, SOAPAttachment2, and so on. These nodes must have a ContentID attribute specified, as well.

4. Sends the SOAP message to a receiver using HTTP.
5. Receives a response.
6. Acknowledges a response.
7. Provides a response to another business process in the application.

Before you use the SendSOAPRequest business process, you must complete the following tasks:

- ◆ Create a SOAP Outbound service configuration with a name and description.
- ◆ Create a separate business process for each SOAP service you want the application to call. The business process must complete the following tasks:
 - ◆ Assign a value to SOAPRequestURL. The value is the URL to which the application should post the SOAP request.
 - ◆ Perform an operation with InvokeBusinessProcessService as the participant, assigning SendSOAPRequest to the WDF_NAME parameter.
- ◆ Depending on your needs, assign values to other parameters before using the SendSOAPRequest business process.

The following table identifies parameters for various SendSOAP functions and the values you may need to specify in your business process:

Required SOAP Function	SOAP Parameter	Parameter Value
SOAPAction HTTP header must be present	SOAP_ACTION	HTTP header
SOAP enveloping on primary document before posting	ADD_SOAP_ENVELOPING	true – Add SOAP enveloping (default). false – No SOAP enveloping.
Digitally sign a SOAP message	SIGN_WITH_KEY	Exact name of the digital certificate.
Envelope namespace prefix for SOAP message	SOAPEnvNSPrefix	Envelope namespace prefix. ADD_SOAP_ENVELOPING must also be true. The default is SOAP_ENV.
Distinguish whether expected response is a SOAP message	SOAP_RESPONSE_NON_SOAP	true – Response is not a SOAP message. false – Response is a SOAP message. Default.
Verify a SOAP response that has been digitally signed	VERIFY_WITH_KEY	Name of the public key. Specify this value before you use the SendSOAPRequest business process if you expect a signature on the SOAP response.
Must use a proxy server	xport-tp-proxyauth	The following information in this format: <i>IP address,port number,username,password</i> User name and password may not be required.

Required SOAP Function	SOAP Parameter	Parameter Value
Envelope namespace for SOAP message	SOAPEnvNSURI	Envelope namespace. The default is: http://www.w3.org/2001/06/soap-envelope

Business Process Example – SendSOAPRequest

The following is the predefined SendSOAPRequest business process, as delivered with the application:

```
<process name="SendSOAPRequest">
  <sequence>

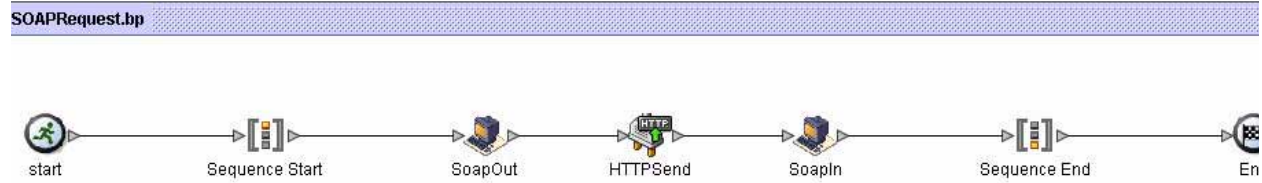
    <operation name="SoapOut">
      <participant name="SOAPOutbound"/>
      <output message="BPML:output">
        <assign to="." from="*"></assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="HTTPSend">
      <participant name="SOAP_HTTP_Send"/>
      <output message="Xout">
        <assign to="b2b-raw-response">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="SoapIn">
      <participant name="SOAPInbound"/>
      <output message="BPML:output">
        <assign to="." from="*"></assign>
        <assign to="bootstrap">>false</assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

The following illustration shows the business process as it appears in the GPM:



Handling Multiple Incoming Attachments

When the SOAP Inbound Service handles a SOAP request containing attachments, the message package is opened and the SOAP payload and all attachments are placed in Process Data. The SOAP payload becomes the primary document, and the attachments are added with keys being SOAPAttachment1 . . . SOAPAttachment*n*. These attachments are located under a SOAPAttachments tag in process data, and their nodes will contain attributes for Content Location and/or Content ID, if these values exist in the SOAP message package. An example Process Data section is shown below for a SOAP message containing two attachments:

```

<ProcessData>
...
<SOAP_URI>/ws/soap/testattachments </SOAP_URI>
<SOAP_INTERMEDIATE_NODE>>false </SOAP_INTERMEDIATE_NODE>
<WFD_NAME>ExtractSOAPAttachments </WFD_NAME>
<SOAPAttachments>
  <SOAPAttachment1 Content-ID='cid:http://productheaven.com/234sdaim3324.xml'
Content-Location='http://productheaven.com/234sdaim3324.xml'
SCIObjctID='neith:4aedaa:eb9e01efbe:-7d3e' />
  <SOAPAttachment2 Content-ID='cid:http://productheaven.com/3874923874.xml'
Content-Location='http://productheaven.com/3874923874.xml'
SCIObjctID='neith:4aedaa:eb9e01efbe:-7d3c' />
</SOAPAttachments>
<SOAP-ENV:ReceivedSOAPHeaders>
  <abc:Extension2 s:mustUnderstand='0'
xmlns:abc='http://example.org/2001/06/ext' />
</SOAP-ENV:ReceivedSOAPHeaders>
<PrimaryDocument SCIObjctID='neith:4aedaa:eb9e01efbe:-7d3a' />
</ProcessData>
  
```

Accessing the SOAP Payload and Attachments in Process Data.

The SOAP payload is the primary document, but to work on an attached document that is referenced by the primary document, additional steps must be taken. Essentially, we want to make the referenced attachment the new primary document. See the following example SOAP payload:

```

<Products>
  <product href="cid:http://productheaven.com/234sdaim3324.xml" />
  <product href="cid:http://productheaven.com/3874923874.xml" />
</Products>
  
```

First, use the XML Encoder service to put the actual contents of the soap payload into process data. Then use an assign statement to make a referenced attachment the new primary document.

Note: You may also want to save the existing primary document for later access. That can be done with an assign such as `<assign to="OldPrimary" from="/ProcessData/PrimaryDocument" />`

The xpath expression would be:

```
<assign to="PrimaryDocument" from="SOAPAttachments/*[@Content-ID=//product[1]/@href/text()]/@SCIOBJECTID" />
```

This xpath expression says to set the primary document to the node under SOAPAttachments having a content ID attribute value that matches the href attribute value of the first occurrence of “product” in process data. You can perform whatever operations are needed on that attachment.

To make the old primary document the current primary document again, use a statement such as:

```
<assign to="." from="/ProcessData/OldPrimary/PrimaryDocument" append="false" />
```

SOAP Outbound Service

The following table provides an overview of the Simple Object Access Protocol (SOAP) Outbound service:

System Name	SOAPOutbound
Graphical Process Modeler (GPM) categories)	All Services, Internet B2B > SOAP
Description	<p>SOAP is an XML-based protocol that enables the application to access services, objects, and servers in a platform-independent manner.</p> <p>The SOAP Outbound Service is responsible for preparing SOAP requests and responses to be sent. This may include adding SOAP enveloping to an XML document, creating a SOAP fault message, and adding SOAP header blocks from process data. The resulting document will either be the raw SOAP request or raw SOAP response, and a B2B HTTP Client adapter configuration sends the document to the necessary location.</p>
Business usage	Prepares SOAP requests and responses.
Usage example	See the predefined business process, SendSOAPRequest
Preconfigured?	Yes. A configuration of the service called SOAPOutbound is installed with the application. No additional setup is needed.
Requires third party files?	No
Platform availability	All supported application platforms
Related services	SOAP Inbound service, HTTP Server adapter, HTTP Client adapter Note: HTTP Server adapter and the HTTP client adapter replace the B2B HTTP Server adapter and HTTP Send service, which have entered the retirement process. For more information, see <i>Retiring Services and Adapters</i> .
Application requirements	None
Initiates business processes?	No
Invocation	If this service is used to send a SOAP response, the parameter SOAP_MODE should be set to respond . The default value for this parameter is send .
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – The output document was created successfully.◆ Error – Errors were encountered somewhere during the creation of the SOAP request or response.
Restrictions	None
Testing considerations	Use the predefined business process, SendSOAPRequest, to test the service.

How the SOAP Outbound Service Works

The application determines whether the SOAP Outbound service should be run in send or respond mode. The default mode is send.

When the service runs in send mode, the application does the following:

- ◆ Adds SOAP enveloping, including header blocks, if necessary.
- ◆ Generates the necessary HTTP headers.
- ◆ Sets some values to be used by the HTTP Client adapter.

When the service runs in respond mode, the application does the following:

- ◆ Checks for the presence SOAPFault in process data, and if it exists, uses the SOAP fault data to generate a SOAP fault.
- ◆ If there is no fault and the application is not in intermediate mode, it adds SOAP enveloping, including header blocks, to the primary document.
- ◆ Generates the HTTP response code header.

Additional Functions

The SOAP Outbound service supports two additional, optional parameters: SOAPEnvAttribute and SOAPAddXMLDeclaration. Both can be passed to the service in an XML business process. The parameters are described in the following table:

Parameter	Description
Add Custom Attributes to the Envelope Element	You can add custom attributes to the Envelope element in a generated outbound SOAP message. The SOAP Outbound service looks for a parameter called SOAPEnvAttribute. The value of this parameter contains the complete attribute specification (of the form attributeName="attributeValue"). You can use this parameter multiple times with append="true" in the Assign statement.
Include an XML Declaration in Generated XML Documents	To include an XML declaration in XML documents generated by the SOAP Outbound service, set the parameter "SOAPAddXMLDeclaration" to true or yes.

SendSOAPRequest Business Process

The SendSOAPRequest business process enables the application to send SOAP messages.

When the application sends a SOAP message, SendSOAPRequest completes the following process:

1. Applies a SOAP envelope to the data, if requested.
2. Adds SOAP headers to an outbound request if the process data contains SOAP headers. The business process takes all subordinate nodes under the SOAPHeaders node in process data and adds them as SOAP header blocks.
3. Adds SOAP attachments to an outbound request if the process data contains SOAP outbound attachments. See <Hyperjump>Sending Multiple Attachments.
4. Sends the SOAP message to a receiver using HTTP.

5. Receives a response.
6. Acknowledges a response.
7. Provides a response to another business process in the application.

Before you use the SendSOAPRequest business process, you must complete the following tasks:

- ◆ Create a SOAP Outbound service configuration with a name and description.
- ◆ Create a separate business process for each SOAP service you want the application to call.

The business process must complete the following tasks:

- ◆ Assign a value to SOAPRequestURL. The value is the URL to which the application should post the SOAP request.
- ◆ Perform an operation with InvokeBusinessProcessService as the participant, assigning SendSOAPRequest to the WDF_NAME parameter.
- ◆ Depending on your needs, assign values to other parameters before using the SendSOAPRequest business process.

The following table identifies parameters for various SendSOAP functions and the values you may need to specify in your business process:

Required SOAP Function	SOAP Parameter	Parameter Value
SOAPAction HTTP header must be present	SOAP_ACTION	HTTP header
SOAP enveloping on primary document before posting	ADD_SOAP_ENVELOPING	true – Add SOAP enveloping (default) false– No SOAP enveloping
Digitally sign a SOAP message	SIGN_WITH_KEY	Exact name of the digital certificate.
Envelope namespace prefix for SOAP message	SOAPEnvNSPrefix	Envelope namespace prefix. ADD_SOAP_ENVELOPING must also be true. The default is SOAP_ENV.
Distinguish whether expected response is a SOAP message	SOAP_RESPONSE_NON_SOAP	true – Response is not a SOAP message. false – Response is a SOAP message. Default.
Verify a SOAP response that has been digitally signed	VERIFY_WITH_KEY	Name of the public key. Specify this value before you use the SendSOAPRequest business process if you expect a signature on the SOAP response.
Must use a proxy server	xport-tp-proxyauth	The following information in this format: <i>IP address,port number,username,password</i> . User name and password may not be required.
Envelope namespace for SOAP message	SOAPEnvNSURI	Envelope namespace. The default is http://www.w3.org/2001/06/soap-envelope .

Business Process Example – SendSOAPRequest

The following is the predefined SendSOAPRequest business process, as delivered with the application:

```
<process name="SendSOAPRequest">
  <sequence>

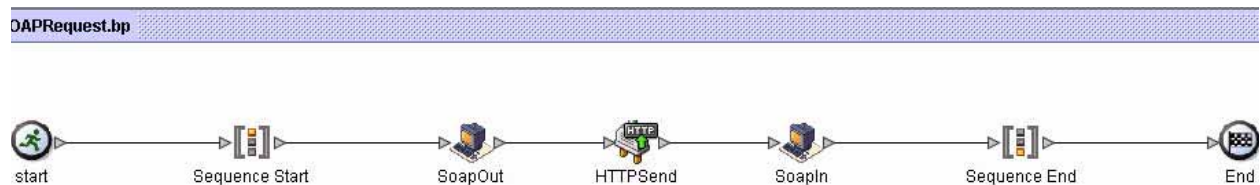
    <operation name="SoapOut">
      <participant name="SOAPOutbound"/>
      <output message="BPML:output">
        <assign to="." from="*"></assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="HTTPSend">
      <participant name="SOAP_HTTP_Send"/>
      <output message="Xout">
        <assign to="b2b-raw-response">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="SoapIn">
      <participant name="SOAPInbound"/>
      <output message="BPML:output">
        <assign to="." from="*"></assign>
        <assign to="bootstrap">>false</assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

The following illustration shows the business process as it appears in the GPM:



Sending Multiple Attachments

The SOAP Outbound service will prepare a SOAP message package (SOAP document with attachments(s)), if desired. It uses the primary document (or the entire SOAP message, depending on configuration) and looks for attachments in the SOAP payload in process data in ProcessData under SOAPOutboundAttachments. The following are required:

- ◆ Documents 1 through n must be named SOAPOutboundAttachments/SOAPAttachment1 through SOAPOutboundAttachments/ SOAPAttachment n
- ◆ A unique Content-ID must be set for any attachments. This can be done with an assign statement such as the following:

```
<assign  
to="SOAPOutboundAttachments/SOAPAttachment1/@Content-ID">http://www.stercomm.com/mya  
ttachment1.txt</assign>
```

Sleep Service

The Sleep service pauses a business process for the amount of time specified in its configuration. The following table provides an overview of the Sleep service:

System name	SleepService
Graphical Process Modeler (GPM) categories	All Services, System
Description	Sleeps for the amount of time specified in the business process.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	Success, Error
Restrictions	No
Persistence level	None

Configuring the Sleep Service

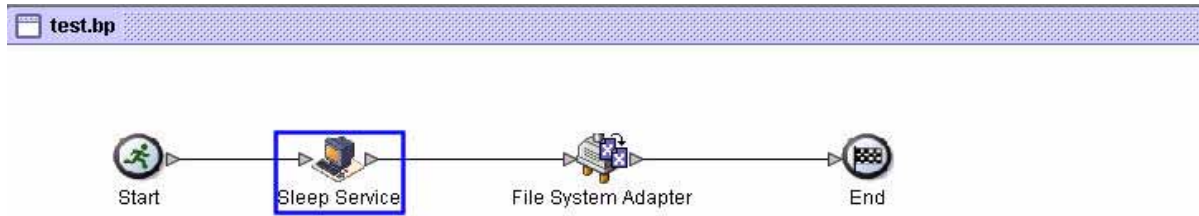
Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
MSEC_SLEEP_INTERVAL	Duration of sleep time in milliseconds. Required.
SLEEP_INTERVAL	Duration of sleep time in seconds. Required.

Business Process Example

The following example illustrates how the Sleep service can be used in a business process. The first service in the business process is the Sleep service, which causes the business process to pause for 60 seconds, then uses the File System adapter to collect a file from a folder called “Test” on the local drive:

This is how the business process looks in the GPM:



Service Editor - Sleep Service

Name: Sleep Service

Config: SleepService

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data

Message Name: SleepServiceInputMessage

Name	Value	Use XPATH?	Append?
MSEC_SLEEP_INTERVAL		<input type="checkbox"/>	<input type="checkbox"/>
SLEEP_INTERVAL	60	<input type="checkbox"/>	<input type="checkbox"/>

This is the BPML for the business process:

```

<process name="default">
  <sequence>
    <operation name="Sleep Service">
      <participant name="SleepService"/>
      <output message="SleepServiceInputMessage">
        <assign to="SLEEP_INTERVAL">60</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="File System Adapter">
      <participant name="Test"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
  
```

This is the status message for the Sleep service step, which you can view from the Execution Manager after the business process completes:



Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the Sleep service:

Parameter	Description
MSEC_SLEEP_INTERVAL	Duration of sleep time in milliseconds.
SLEEP_INTERVAL	Duration of sleep time in seconds.

Sterling Control Center Service

The following table provides an overview of the Sterling Control Center service:

System Name	Sterling Control Center Service
Graphical Process Modeler (GPM) category	Applications > Sterling Commerce > Sterling Control Center
Description	This service handles requests from the Sterling Control Center. The types of requests include file transfer activity history, business process history, perimeter service status, and adapter status.
Business usage	Application uses this service to communicate with the Sterling Control Center to adhere to specific service level agreements and to be notified in advance if conformance to service level windows is in jeopardy. Application also uses this service to be alerted about file transfer activity, business process activity, and different adapter status in general.
Usage example	A trading partner attempts to transfer a file, but the connection drops and the transfer does not complete. This causes a visibility event to be generated in Application. The Sterling Control Center receives the information about the event through a SOAP request to the Sterling Control Center service in Application. The Sterling Control Center generates an alert because the configured service level criteria have been missed. Appropriate parties are notified and mitigating action can be taken.
Preconfigured?	Yes
Requires third party files?	No third party files are required.
Platform availability	All supported Application platforms
Related services	None
Application requirements	Used at a system level by the Sterling Control Center product. Must be referenced in the definition of a Application Web Services Manager configuration.
Initiates business processes?	This service does not initiate business processes.
Invocation	This service is invoked from a Web service.
Business process context considerations	None
Returned status values	Possible status values that can be returned from this service are: <ul style="list-style-type: none">◆ 0 - Success◆ 1 - Error
Restrictions	None
Persistence level	None
Testing considerations	Debug information for this service can be found in the Sterling Control Center service log files.

Implementing the Sterling Control Center Service

This service is used at a system level by the Sterling Control Center product. It must be referenced in the definition of a Application Web Services Manager configuration.

Configuring the Sterling Control Center Service

The Sterling Control Center service is configured upon installation of Application. Settings for the following fields are provided for reference:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	<p>Group of services or services of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this service. Valid values are:</p> <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time (default).◆ Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>For more information about service groups see <i>Managing Services and Adapters</i>.</p>

Parameters Passed from the Business Process to Service

The following table describes the parameters passed from the business process to the Sterling Control Center service:

Field	Description
RequestType	<p>Request type sent from the Sterling Control Center to the Sterling Control Center service. Required. Valid values are:</p> <ul style="list-style-type: none">◆ getHistory◆ getBPHistory◆ getInfo◆ getStatus◆ getLicense◆ OpsCommand◆ runSql

Field	Description
FromDate	A date and time value that specifies when to get business process or AFT activity history. Required if Request Type = getHistory or getBPHistory.
ProtocolList	A comma-delimited list of protocol activities for the Control Center to monitor. Valid values include: <ul style="list-style-type: none"> ◆ FTP ◆ Connect:Direct ◆ SFTP ◆ HTTP ◆ AS2 ◆ WebDAV Optional. Valid if Request Type = getHistory.
RecordLimit	Specifies the total number of activity history records to return to Control Center in a single request from Control Center. Optional. Valid if Request Type = getHistory or getBPHistory. Valid value is any number less than or equal to 5000. Default is 5000.
Command	The Ops Command to execute in Application. Required if Request Type = Command. Valid values are: <ul style="list-style-type: none"> ◆ ADAPTERSTATUS ◆ consolidatedstatus ◆ LISTADAPTERS ◆ LISTPERIMETERCLIENTS
Sql	SQL to execute in Application. Required if Request Type = Sql. Valid value is any valid SQL. For example, an SQL to get all the business process names.
BpExclusionList	A list of BPs for the Control Center to not collect activity history. Required if Request Type is getBPHistory. Valid value is any business process name. Default is SOA_RequestHandler.

Parameters Passed from Service to Business Process

The following table describes the parameters passed from the Sterling Control Center service to the business process:

Parameter	Description
PrimaryDocument	A primary document is created to return the results of the requests. The contents of the primary document vary based of the request type.

Sterling Information Broker Adapter

The following table provides an overview of the Sterling Information Broker adapter:

System name	SIB_ADAPTER
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Sterling Information Broker
Description	<p>Enables you and your trading partners to exchange business documents (such as purchase orders and invoices) electronically. The Sterling Information Broker adapter provides a single point of connection for business exchanges with many trading partners, regardless of the type of communications protocol or data format. The Sterling Information Broker adapter is a bridge between the many systems and technologies used by various members of the business community:</p> <ul style="list-style-type: none">◆ A bridge between communications protocols, including legacy-based protocols and Internet-based protocols.◆ A bridge between business standards, including translations between traditional electronic data interchange (EDI) standards and emerging Internet standards.◆ A bridge between e-commerce exchanges, including connections to other VANs and connections between traditional VANs and Web-based e-marketplaces.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No

Initiates business processes?	No
Invocation	Runs as part of a business process.

How the Sterling Information Broker Adapter Works

For example, consider the following scenario. A buyer uses FTP protocol and wants to send purchase orders to two suppliers. One supplier resides on XYZ VAN and uses bisynchronous protocol. The other supplier resides on the Internet and uses AS1. The buyer connects once to the Sterling Information Broker, transmits the documents through FTP protocol, and the Sterling Information Broker takes care of the rest. One document is routed to XYZ VAN using bisynchronous protocol. The other document is delivered over the Internet using AS1. The process happens in reverse when the two suppliers return purchase order acknowledgments to the buyer.

The Sterling Information Broker adapter enables Application to communicate with the Sterling Information Broker network using the File Transfer Protocol (FTP). The Sterling Information Broker adapter issues a PUT request to the Sterling Information Broker network to send the document.

The Sterling Information Broker adapter can send documents to and receive documents from the Sterling Information Broker network, and can start from only a business process.

Implementing the Sterling Information Broker Adapter

To implement the Sterling Information Broker adapter, complete the following tasks:

1. Activate your license for the Sterling Information Broker adapter.
2. Create a Sterling Information Broker adapter configuration.
3. Configure the adapter.
4. Set CACert or Keycert in the GPM. For information, see *Before Using the Sterling Information Broker Adapter in a Business Process*.
5. Create and enable a business process that includes the Sterling Information Broker adapter.
6. Test the business process and the adapter.
7. Run the business process.

Configuring the Sterling Information Broker Adapter

To configure the Sterling Information Broker Adapter, you must specify field settings in Application and in the Graphical Process Modeler (GPM).

Application Configuration

The following table describes the fields used to configure the Sterling Information Broker adapter in Application:

Field	Description
Add	Click the plus sign to add a new remote user. Optional. The configuration can be created without adding any remote users.
Username	Name of the remote user. Required.
Password	Password for the remote user. Optional.
Passphrase	Passphrase associated with the remote user. Optional. Used only for SSL client authentication.

GPM Configuration

The following table describes the fields used to configure the Sterling Information Broker adapter in the GPM:

Field	Description
Config	Name of the service configuration.
xport-ftp-asciiExt	A list of one or more file extensions, separated by colons (:) that will change the transfer mode to ascii. The default transfer mode is ASCII but if that is changed to BINARY or AUTO, this field can be used to list file extensions that require ASCII transfer. Example value .txt:.text:.html
xport-ftp-conntype	Connection type that specifies how data connections are made. Valid values are: ACTIVE – Sets the server to make the connection. PASSIVE – Sets the adapter to make the connection.
xport-ftp-dataport	Valid port number provided by your network administrator that is not restricted by the firewall. Valid value is any valid data port number not being used by another application running on the system. Valid format is # (number), #.#, a range of numbers, or any of the three formats together.
xport-ftp-dir	Directory name or mailbox ID of an external trading partner.
xport-ftp-document	Document name to receive. Used only in the GET xport-ftp-mode.
xport-ftp-host	IP address or host name of the external trading partner host system. Valid values are valid IP addresses and host names.
xport-ftp-mode	Command used to determine the data flow. Valid values are PUT and GET. <ul style="list-style-type: none"> ◆ PUT – Send data from Application to the network. ◆ GET – Get data from the network.
xport-ftp-port	Port number of the Sterling Information Broker network. Valid value is a valid 2-digit port number.

Field	Description
xport-ftp-transferMode	Select the transfer mode to use. Valid values are: <ul style="list-style-type: none"> ◆ ASCII ◆ BINARY ◆ AUTO Default is ASCII.
xport-tp-authfile	Authentication file containing the user name, password, and passphrase. You must specify the complete path of the authentication file. This field is valid if the SSL option is MUST. Valid value is the file name. Optional.
xport-tp-cakeycert-id	File name that contains the trusted certificate authority public certificate for Application to use with this adapter. Optional.
xport-tp-cipher	Level of encryption Application is to apply to the data that flows through the socket connection. Valid values are STRONG (default), WEAK, and ALL. Optional.
xport-tp-clntdelay	Number of seconds the adapter will wait before trying again to connect to the Sterling Information Broker network. Optional.
xport-tp-clntretries	Number of times the adapter will try to connect to the Sterling Information Broker network. Optional.
xport-tp-ctlport	Valid port number provided by your network administrator that is not restricted by the firewall. Valid value is any valid data port number not being used by another application running on the system. Valid format is # (number), #.#, a range of numbers, or any of the three formats together.
xport-tp-keycert-id	File name that contains the private key and public certificate, which is signed by the network trusted certificate authority. Valid value is a valid file name. Optional.
xport-tp-proxyauth	Information needed to navigate through firewalls. Valid value is an 8-character alphanumeric string. Optional.
xport-tp-proxyretry	Number of times the adapter will try to connect to the proxy gateway. Optional.
xport-tp-resptimeout	Number of seconds given for the network to respond to the request before the connection attempt times out. Valid values are 1-59 for seconds and # *60 for minutes to hours. Optional.
xport-tp-ssloption	SSL flag that turns on the SSL socket negotiation. Valid values are SSL-NONE and SSL_MUST. Optional.
xport-tp-user	Remote account user name. Can be used to override name set in Application interface configuration.

Before Using the Sterling Information Broker Adapter in a Business Process

If the FTP server that is communicating with Application uses Secure Socket Layer (SSL) communication, the CA certificate object ID or system certificate object ID (CACert or Keycert) must be saved before the Sterling Information Broker adapter can be used in a business process.

To set CACert or Keycert in the GPM:

1. Do you want to set CACert or Keycert?

- ◆ To set CACert, from the **Trading Partner** menu, select **Digital Certificates > CA**.
 - ◆ To set Keycert, from the **Trading Partner** menu, select **Digital Certificates > System**.
2. Search for the appropriate certificate name, using the Search or List function.
 3. In the Digital Certificates page, right-click anywhere in the gray area, and select **View Source** from the menu.
Notepad opens. If notepad does not open, right-click again, select **refresh**, and repeat step 3.
 4. From the **Search** menu, select **Find**.
 5. In the **Find what** field, type the name of the certificate as it appears in the list of search results returned in Application.
The certificate name is highlighted.
 6. One or two lines above the certificate name, look for a value similar to this:
`&id=host:4695a6:eca597dbfd:-8000&`
Between the two ampersands (&) is the object ID.
 7. Copy the object ID.
 8. Open the GPM, and select the appropriate adapter configuration.
 9. Right-click the adapter icon, and select **Properties**.
 10. In the Service Editor, specify whether you are setting CACert or Keycert:
 - ◆ If you are setting CACert, paste the object ID in the **xport-tp-cakeycert-id** field.
 - ◆ If you are setting Keycert, paste the object ID in the **xport-tp-keycert-id** field.
- Note:** Separate multiple certificates with a comma (,).

Example Business Processes

The following example retrieves information from a Sterling Information Broker mailbox:

```
<process name = "SibGet">
  <operation name="SIB_ADAPTER">
    <participant name="SIBTest"/>
    <output message="SIB_ADAPTERInputMessage">
      <assign to="xport-ftp-host">ftp.host.companyx.com</assign>
      <assign to="xport-ftp-mode">GET</assign>
      <assign to="xport-ftp-port">21</assign>
      <assign to="xport-ftp-document">Document.Name</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

The following example sends information to a Sterling Information Broker mailbox:

```
<process name = "SibPut">
  <operation name="SIB_ADAPTER">
    <participant name="SIBTest"/>
    <output message="SIB_ADAPTERInputMessage">
      <assign to="xport-ftp-host">ftp.host.companyx.com</assign>
      <assign to="xport-ftp-mode">PUT</assign>
      <assign to="xport-ftp-port">21</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

Straight Through Extraction Service

The Straight Through Extraction Service is used in a business process to specify what data used during process execution should be saved to the Visibility Services fact repository. The following table provides an overview of the Straight Through Extraction service:

System Name	Straight Through Extraction Service
Graphical Process Modeler (GPM) categories)	All Services
Description	The Straight Through Extraction Service is used in a business process to specify what data used during process execution should be saved to the Visibility Services fact repository
Business usage	This service enables data generated within Application processing to be saved in a Visibility Services fact repository.
Usage example	This service allows a Visibility Services developer to specify data in process data that should be saved as Visibility Services facts. Based on the Event Schema ID specified in the service configuration, the specified data at that step in the business process will be saved to the Visibility Services data system.
Preconfigured?	Yes. A configuration of the service called StraightThroughExtraction is installed with Application.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Mapped Extraction service
Application requirements	The fact model referenced by this service must be created and checked in to Application.
Initiates business processes?	Cannot initiate a business process.
Invocation	From a business process.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	None
Testing considerations	Ensure that the Visibility Services listeners are running. See <i>About Event Listeners</i> for more information.

How the Straight Through Extraction Service Works

1. The Visibility Services developer specifies data in process data that should be extracted for Visibility Services use.

- Based on the Event Schema ID specified in the service configuration, the specified data at that step in the business process will be saved to the Visibility Services fact repository.

Implementing the Straight Through Extraction Service

Use the supplied configuration of the service, `StraightThroughExtraction`, in your business processes.

The only configuration required for the service is specifying the Event Schema ID to be used for your business process. You specify this in the GPM.

Configuring the Straight Through Extraction Service

You must specify field settings in the GPM.

Field	Description
Config	Select the name of the service configuration from the list. You can use the supplied configuration, <code>StraightThroughExtraction</code> .
eventschemaid	Event Schema ID

Business Process Example

This sample calls the service configuration `StraightThroughExtraction` with `eventschemaid` “`bi.system.logs`”. The service will create an event using the value of `eventschemaid` as event schema key and all process data under `/Processdata/EventAttributes` as the event attributes and fire the event. The names under `/Processdata/EventAttributes` must satisfy the event schema specification.

```
<process name="sampleStraightThroughExtraction">
  <sequence name="main seq">
    <assign to="EventAttributes/message">this is message</assign>
    <assign to="EventAttributes/throwable">this is throwable</assign>
    <operation name="StraightThroughExtraction">
      <participant name="StraightThroughExtraction"/>
      <output message="StraightThroughExtractionInputMessage">
        <assign to="eventschemaid">bi.system.logs</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Text Document Appender Service

The following table provides an overview of the Text Document Appender service:

System name	Text Document Appender Service
Graphical Process Modeler (GPM) categories	All Services
Description	Use the Text Document Appender service to add a string at the end of a document in a business process.
Business usage	You receive a document from a trading partner or the document is a translated document. You may need to add a string to the document to process it. The Text Document Appender service allows you to add a string to the document.
Usage examples	The document you received from a trading partner does not have a line terminator in the document. The Text Document Appender allows you to add a line terminator which is required for processing the document.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	None
Application requirements	N/A
Initiates business processes?	No
Invocation	This service is added as a step in a business process. When you invoke the business process, the Text Document Appender service adds the specified string at the end of the primary document.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success◆ Error
Restrictions	N/A
Persistence level	System Default
Testing considerations	None

Configuring the Text Document Appender Service

To configure the Text Document Appender service, you must specify settings for the following fields in the GPM:

Field	Description
appendString	Specify the string value to be added at the end of the primary document. Valid value is any string. Required.

Output from Service to Business Process

The following table contains the parameter passed from the Text Document Appender service to the business process:

Parameter	Description
appendString	Specify the string value to be added at the end of the primary document. Valid value is any string. Required.

Output from Business Process to Service

The Text Document Appender service adds the string value you specify in the appendString parameter at the end of the primary document. If no primary document is available, the Text Document Appender service creates a primary document and adds the string value you specify in the appendString parameter to the primary document.

Business Process Example

The following example business process illustrates using the Text Document Appender service:

```
<process name="default">
  <operation name="Text Document Appender">
    <participant name="TextAppender"/>
    <output message="DocAppendInputMessage">
      <assign to="appendString">Append String</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

This Service

The This service is a multi-purpose utility service with the following functions:

- ◆ Obtain different types of information from a business process and its documents.
- ◆ Make this information available to the user in process data.
- ◆ Assign and release (remove) items from process data.

Note: The functionality of the This service is being replaced with other adapters and services designed more specifically to perform the services. The This service will be retired in the future.

Replacement Services

If you are developing a new application, you are strongly encouraged to select one of the replacement services as they are more easily integrated into a business process and provide additional functionality. The following services provide the functionality of the This service, plus additional features:

Service	Function
BP MetaData Info Service	<p>The BP Metadata Info service enables you to retrieve information about a particular business process including:</p> <ul style="list-style-type: none">◆ Business process definition ID◆ Persistence level◆ Lifespan◆ Business process definition version◆ Sub/parent business process information
Get Document Info Service	<p>The Get Document Info service provides information about the primary document. You can query the service about information including the following:</p> <ul style="list-style-type: none">◆ Document Name◆ Document Body Length◆ Document ID◆ Document Subject◆ Document Create Time◆ Document Content Type◆ Document Char Encoding◆ Document Content Subtype◆ Previous Document ID◆ Document Length
Release Service	<p>The Release service is a system service used to discard previous assignments in process data.</p>

Service	Function
TimeStamp Utility Service	<p>The TimeStamp Utility Service enables you to create, format, and perform arithmetic operations on timestamp values in a business process. The following options are available with the Timestamp Utility service:</p> <ul style="list-style-type: none"> ◆ Create a timestamp for the current time. ◆ Add the numeric values of two timestamps. ◆ Subtract one timestamp value from another. ◆ Format timestamp values.

This Service Overview

The following table provides a high-level overview of the This service:

System name	This
Graphical Process Modeler (GPM) categories	All Services, System
Description	A multi-purpose utility service that obtains a variety of information from a business process and its documents and makes this information available in process data. It also assigns and releases items from process data.
Business usage	<p>Perform the following utility functions within the business process:</p> <ul style="list-style-type: none"> ◆ Obtain the ID of the current business process. ◆ Obtain document metadata (name, document ID) for future use in the business process. ◆ Load a document or information into process data. ◆ Release a node from process data. ◆ Make an assignment to process data. ◆ Obtain a timestamp on the process. ◆ Obtain the elapsed time between a start time and a finish time.
Usage example	The This service can be used in a variety of ways including getting time and date information for a business process, retrieving information about a document including the name and document ID, and loading and releasing information from process data.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms

Related services	The functions available in the This service are being replaced by other adapters and services, including: <ul style="list-style-type: none"> ◆ Release Service ◆ TimeStamp Utility Service ◆ Get Document Info Service ◆ BP MetaData Info Service
Application requirements	None
Initiates business processes?	None
Invocation	Runs as part of a business process.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none"> ◆ Success: If the operation was successful ◆ Failure: Otherwise
Restrictions	None
Persistence level	None
Testing considerations	Verify that you have an existing preconfigured This service. Create and test a business process that uses the This service and the desired functions.

How the This Service Works

The This service can be used to access information and make it available to a business process in the process data and to remove specific items from process data.

Examples of all functions are included in *Usage Examples*.

Implementing the This Service

When implementing the This service for use in a business process, This service is normally preconfigured and available after Application installation. If there is not an existing service configuration, then create a service configuration.

GPM Configuration and BPML Parameters

All *Message To Service* parameters for the This service are entered in the GPM using the Advanced Editor on the Message To tab. *Message From Service* parameters are entered by clicking Add on the Message From tab.

Input Parameters to the This Service

The following table describes the parameters input from the business process to the This service. These values are either coded in the BPML or set in the GPM configuration of the service. The BPML element values are provided in parentheses for reference.

Parameter (BPML Element Value)	Description
Document node name (documentKey)	The name of the node for the document in process data. Required for the takeDocument and giveDocument functions.
Document Identifier (documentId)	The ID assigned by Application for the document to load. Required for the takeDocument function.
Source Node (from)	The node in process data to release or assign from. Required for assign and release.
Target Node (to)	The node in process data to assign to. Required for assign.
Append flag (append)	Replace duplicate assignment in process data. Optional. Valid values: <ul style="list-style-type: none"> ◆ True – The value for the duplicate tag in process data is replaced. ◆ False – A new entry for the duplicate tag is added to process data. Default.
Trim whitespace flag (trimWhitespace)	Delete whitespace before the assignment is made. Optional. Valid values are: <ul style="list-style-type: none"> ◆ True – Any whitespace in the assignment to process data is deleted. ◆ False – No whitespace is deleted. Default.
Time Format syntax (format)	The syntax to format a millisecond timestamp. The standard 1.3.1 Java object SimpleDate format is used to define the syntax (see Javadoc http://java.sun.com/j2se/1.3.1/docs/api/index.html for details). Optional. Default format is yyyy-MM-dd'T'HH:mm:ss.
Universal Time flag (isTimeUTC)	The timestamp is in UTC. Optional. Valid values are: <ul style="list-style-type: none"> ◆ True – UTC ◆ False – GMT. Default
Start time (startTimeMillis)	Start timestamp in milliseconds. This is the start time for the timeDiffRequest function. Required (timeDiff).
End time (endTimeMillis)	End timestamp in milliseconds. This is the end time for the timeDiffRequest function. Required (timeDiff).

Output Parameters to the Business Process/Process Data

The following table describes the parameters that are output from the This service to the business process. The term *document* in the table refers to the document that the This service is evaluating, either the primary

document or another document in process data. The BPML element values are provided in parentheses for reference.

Parameter (BPML Element Value)	Description
Document Identifier (documentId)	The Document ID assigned by Application for the document.
Document Name (documentName)	The bodyName (file name) from the document. The file name is the name that is associated with the document, either the name of the file from the file system or the name assigned to the file by Application.
Timestamp (time)	Current timestamp.
Timestamp difference (elapsedTimeMillis)	The elapsed time in milliseconds between the start time and end time entered as input to the This service.

Output Messages

The name of the output message on Message To Service (message input to the This service) defines the functional behavior for the service. The function name is either typed in the Message Name field of the Message To tab in the Service Editor of the GPM or is coded in BPML as the output message (<output message="*functionname*">). The following message types are supported:

- ◆ `thisRequest` – Obtains the business process ID.
- ◆ `giveDocumentRequest` – Obtains the doc ID for the input document.
- ◆ `takeDocumentRequest` – Adds the document to process data.
- ◆ `getDocumentName` – Obtains the name of the document.
- ◆ `releaseRequest` – Releases the selected node from process data.
- ◆ `assignRequest` – Adds the requested item to process data.
- ◆ `timeRequest` – Obtains the current timestamp.
- ◆ `timeDiffRequest` – Obtains the elapsed time in milliseconds between two timestamps.

Usage Examples

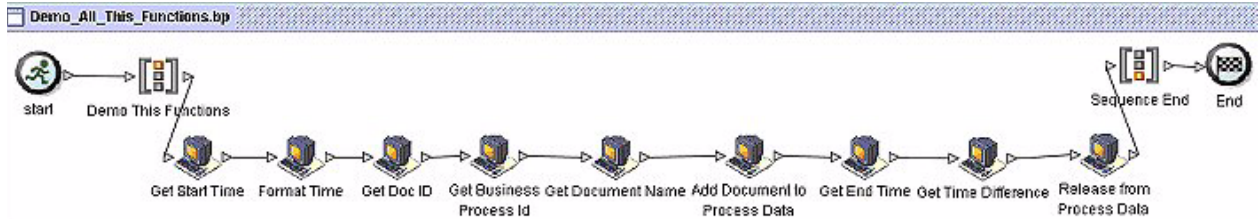
The following examples illustrate how the different functions available in the This service might be implemented in a business process.

This business process does the following:

1. Gets the current (start) time and assigns it to tag <time1> in process data.
2. Gets the current time and formats it in process data.
3. Gets the document ID of the primary document and places it in process data.
4. Gets the business process ID and places it in process data.
5. Gets the name of the primary document and places it in process data.

6. Takes the primary document, assigns it to Document1 and makes a reference to it available in process data.
7. Gets the current time and assigns it to tag <time2> in process data.
8. Gets the time difference in milliseconds between time2 and time1 and assigns it to a variable in process data.
9. Releases the primary document from process data.

This is the business process as shown in the GPM:



Each of the icons is a configuration of the This service configured to perform the actions that the name describes.

The following demonstrates the configuration for each of the This service functions in the GPM and the corresponding BPML. Process data is also shown for each example to demonstrate what is output to process data from the service.

Note: All GPM parameters are assigned using either the Advanced Editor for the Message From Service tab or the Add button for the Message To Service tab.

Get Start Time and Get End Time (timeRequest function)

The Get Start Time and Get End Time configurations put the current time in process data. The Get Start Time and Get End Time configurations are the same except that the parameter name assigned to process data is time1 for the start time and time2 for the end time.

Set the following parameters in the GPM Message From Service tab:

1. Select **Allow message write** from the Input Msg drop-down list and click **Add**.

2. In the Advanced Editor, assign * (asterisk) to **time1**, select the Use XPATH? check box, and click **OK**.

Service Editor-Get Start Time

Name: Get Start Time

Config: This

Message To Service | Message From Service

Input Msg: Allow message write

Name	Value	Use XPATH?
time1	*	<input checked="" type="checkbox"/>

Set the following parameters in the GPM Message To Service tab:

1. Select **Messages Only** from the Output Msg drop-down list.
2. Set Message Name to **timeRequest**.

Service Editor-Get Start Time

Name: Get Start Time

Config: This

Message To Service | Message From Service

Output Msg: Messages Only

Message Name: timeRequest

The following illustration shows the BPML:

```

<operation name = "Get Start Time">
  <participant name="This"/>
  <output message="timeRequest">
  </output>
  <input message="inmsg">
    <assign to="time1" from="*" />
  </input>
</operation>

```

Function used by the This Service to obtain the current time

The timeRequest output is placed in process data within the <time1> tags. The <time> tag contains the time stamp that can be formatted. The <currentTimeMillis> tag always contains the time in milliseconds:


```

<time1>
  <currentTimeMillis>1077632480082</currentTimeMillis>
  <time>1077632480082</time>
</time1>

```

Format Time (timeRequest function)

The Format Time Configurations of the This service uses the timeRequest function and puts the formatted current time in process data. The format of the date is specified in the Message To Service format parameter. Format values used are the Java Simple Date Formats.

Note: For more information about the SimpleDateFormat, see <http://java.sun.com/j2se/1.3/docs/api/java/text/SimpleDateFormat.html>.

Set the following parameters in the GPM Message To Service tab:

1. Select **Messages Only** from the Output Message drop-down list.
2. Set Message Name to **timeRequest** and click **Add**.
3. In the Advanced Editor, assign **yyyy-MM-dd:HH:mm:ss** to **format** and click **OK**.

Name	Value	Use XPATH?
format	yyyy-MM-dd:HH:mm:ss	<input type="checkbox"/>

Set the following parameters in the GPM Message From Service tab:

1. Select **Allow message write** from the Input Msg drop-down list and click **Add**.
2. In the Advanced Editor, assign * (asterisk) to **time1** and click **OK**.

The following illustration shows the BPML:

```
<operation name="Format Time">
  <participant name="This"/>
  <output message="timeRequest">
    <assign to="format">yyyy-MM-dd:HH:mm:ss</assign>
  </output>
  <input message="inmsg">
    <assign to="timestamp" from="time"/>
  </input>
</operation>
```

The date format is assigned to "format" in the output message.

The formatted time output is placed in process data within the <timestamp> tags:

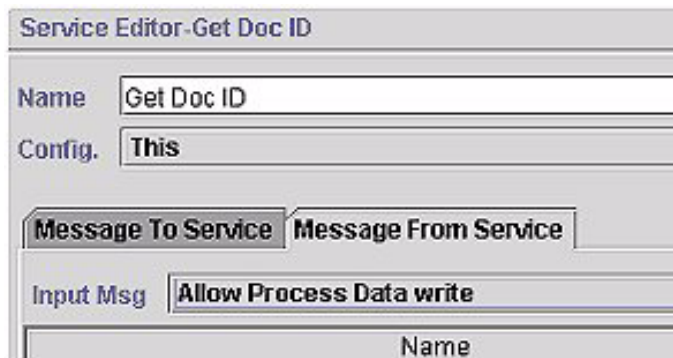
```
<timestamp>
  <time>2004-02-24:07:21:20</time>
</timestamp>
```

Get Doc ID (giveDocumentRequest function)

The Get Doc ID Configuration of the This service uses the giveDocumentRequest function. It gets the ID of the document (in this case the primary document) and puts it in process data.

Set following parameter in the GPM Message From Service tab:

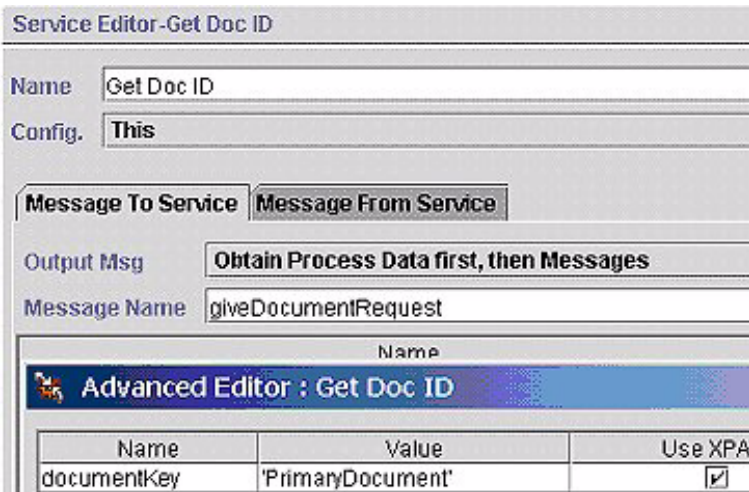
- ◆ Select **Allow Process Data write** from the Input Msg drop-down list.



Set the following parameters in the GPM Message To Service tab:

1. Select **Obtain process Data first, then Messages** from the Output Msg drop-down list.
2. Set Message Name to **giveDocumentRequest** and click **Add**.

3. In the Advanced Editor, assign **PrimaryDocument** to **documentKey**, select the Use XPATH? check box, and click **OK**.



The following illustration shows the BPML:

```

<operation name = "Get Doc ID">
  <participant name="This" />
  <output message="giveDocumentRequest">
<assign to="." from="*">
</assign>
  <assign to="documentKey" from="'PrimaryDocument'"/>
  </output>
  <input message="inmsg">
<assign to="." from="*" />
  </input>
</operation>

```

Output message that is required for the service to obtain the document ID

The document ID output is placed in the process data within the <documentId> tags:

```
<documentId>L2000-000248:f96c:fae7d336aa:-784c</documentId>
```

Get Business Process ID (thisRequest function)

The Get Business Process ID configuration of the This service uses the thisRequest function. It gets the process ID for the business process and places it in process data.

Set the following parameters in the GPM Message From Service tab:

1. Select **Allow message write** from the Input Msg drop-down list and click **Add**.

2. In the Advanced Editor, assign **this/node()** to **thisProcessInstance**, select the Use XPATH? check box, and click **OK**.

The screenshot shows the 'Service Editor - Get Business Process Id' window. The 'Name' field is 'Get Business Process Id' and the 'Config.' field is 'This'. The 'Message To Service' tab is selected. The 'Input Msg' dropdown is set to 'Allow message write'. Below this is a table with three columns: 'Name', 'Value', and 'Use XPATH?'. The table contains one row with 'thisProcessInstance' in the Name column, 'this/node()' in the Value column, and a checked checkbox in the Use XPATH? column.

Name	Value	Use XPATH?
thisProcessInstance	this/node()	<input checked="" type="checkbox"/>

Set the following parameters in the GPM Message To Service tab:

1. Select **Messages only** from the Output Msg drop-down list.
2. Set Message Name to **thisRequest**.

The screenshot shows the 'Service Editor - Get Business Process Id' window. The 'Name' field is 'Get Business Process Id' and the 'Config.' field is 'This'. The 'Message To Service' tab is selected. The 'Output Msg' dropdown is set to 'Messages Only'. The 'Message Name' field is set to 'thisRequest'.

The following illustration shows the BPML:

```
<operation name="Get Business Process Id">
  <participant name="This"/>
  <output message="thisRequest"/>
  <input message="inmsg">
    <assign to="thisProcessInstance" from="this/node()" />
  </input>
</operation>
```

The business process ID output is placed in process data within the `<thisProcessInstance><INVOKE_ID_LIST>` tags.

```
<thisProcessInstance>
  <INVOKE_ID_LIST>27031</INVOKE_ID_LIST>
</thisProcessInstance>
```

Get Document Name (getDocumentName)

The Get Document Name configuration of the This service gets the name of the primary document and places it in process data.

Set the following parameter in the GPM Message From Service tab:

- ◆ Select **Allow Process Data write** from the Input Msg drop-down list.

Service Editor - Get Document Name

Name: Get Document Name

Config: This

Message To Service | **Message From Service**

Input Msg: **Allow Process Data write**

Name

Set the following parameters in the GPM Message To Service tab:

1. Set Message Name to **getDocumentName**.
2. Select **Obtain Process Data first, then Messages** from the Output Msg drop-down list and click **Add**.
3. In the Advanced Editor, assign **PrimaryDocument** to **documentName**, select the Use XPATH? check box, and click **OK**.

Service Editor - Get Document Name

Name: Get Document Name

Config: This

Message To Service | Message From Service

Output Msg: **Obtain Process Data first, then Messages**

Message Name: getDocumentName

Name

Advanced Editor : Get Document Name

Name	Value	Use XPATH?
documentName	'PrimaryDocument'	<input checked="" type="checkbox"/>

The following illustration shows the BPML:

```
<operation name = "Get Document Name">
  <participant name="This"/>
  <output message="getDocumentName">
    <assign to="." from="*">
</assign>
  <assign to="documentName" from="'PrimaryDocument' ">
</assign>
</output>
  <input message="inmsg">
    <assign to="." from="*">
</input>
</operation>
```

The output message in BPML is set to getDocumentName.

The name of the document is placed in process data within the <documentName> tags. In this case the document being processed was named SampleDocument.xml:

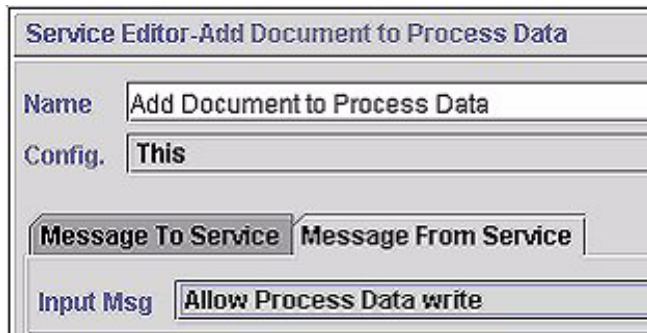
```
<documentName>SampleDocument.xml</documentName>
```

Add Document to Process Data (takeDocumentRequest)

The Add Document to Process Data configuration of the This service adds a document to process data.

Set the following parameter in the GPM Message From Service tab:

- ◆ Select Allow Process Data write from the Input Message drop-down list.



Set the following parameters in the GPM Message To Service tab:

1. Set Message Name to **takeDocumentRequest**.
2. Select **Messages Only** from the Output Msg drop-down list and click **Add**.
3. In the Advanced Editor, assign **documentId/text()** to **documentId** and select the Use XPath? check box. This gets the documentId previously entered in the process data using XPath. See *Get Doc ID (giveDocumentRequest function)*.

- Assign **Document1** to **documentKey** in the Advanced Editor, select the Use XPATH? check box, and click **OK**.

Service Editor-Add Document to Process Data

Name: Add Document to Process Data

Config: This

Message To Service: Message From Service

Output Msg: Messages Only

Message Name: takeDocumentRequest

Name	Value	Use XPATH?
documentId	documentId/text()	<input checked="" type="checkbox"/>
documentKey	'Document1'	<input checked="" type="checkbox"/>

The following illustration shows the BPML:

```
<operation name = "Add Document to Process Data">
  <participant name="This" />
  <output message="takeDocumentRequest">
    <assign to="documentKey" from="'Document1' " />
    <assign to="documentId" from="documentId/text() " />
  </output>
  <input message="inmsg">
    <assign to="." from="*" />
  </input>
</operation>
```

The document with the specified documentId is placed in process data in the Document1 node:

```
<Document1 SCIObjectID="L2000-000248:f96c:fae7d3336aa:-7820" />
```

Get Time Difference (takeDocumentRequest)

The Get Time Difference configuration of the This service takes two timestamps (a start time and an end time) as input and outputs the difference in milliseconds to process data.

Set the following parameters in the GPM Message From Service tab:

- Select **Allow message write** from the Input Msg drop-down list and click **Add**.

- In the Advanced Editor, assign **elapsedTimeMillis** to **diffTimestamp**, select the Use XPATH? check box, and click **OK**.

Service Editor-Get Time Difference

Name: Get Time Difference

Config: This

Message To Service | Message From Service

Input Msg: Allow message write

Name	Value	Use XPATH?
diffTimestamp	elapsedTimeMillis	<input checked="" type="checkbox"/>

Set the following parameters in the GPM Message To Service tab:

- Select **Messages Only** from the Output Msg drop-down list.
- Set Message Name to **timeDiffRequest** and click **Add**.
- In the Advanced Editor, assign **time1/currentTimeMillis/text()** (the start time previously placed in process data) to **startTimeMillis** and select the Use XPATH? check box.
- Assign **time2/currentTimeMillis/text()** (the end time previously placed in process data) to **endTimeMillis**, select the Use XPATH? check box, and click **OK**.

Service Editor-Get Time Difference

Name: Get Time Difference

Config: This

Message To Service | Message From Service

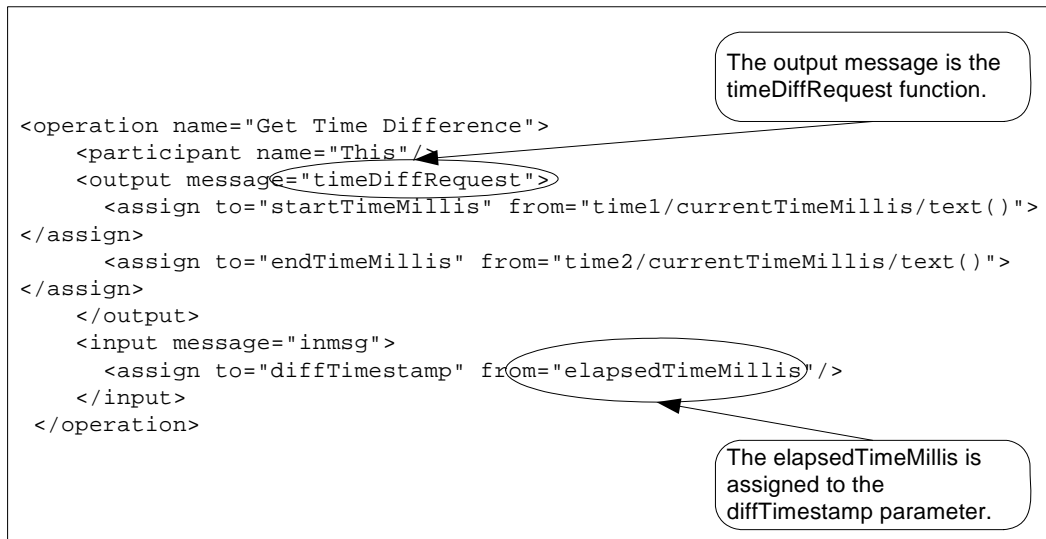
Output Msg: Messages Only

Message Name: timeDiffRequest

Advanced Editor : Get Time Difference

Name	Value	Use XPATH?
endTimeMillis	time2/currentTimeMillis/text()	<input checked="" type="checkbox"/>
startTimeMillis	time1/currentTimeMillis/text()	<input checked="" type="checkbox"/>

The following illustration shows the BPML:



The elapsed time in milliseconds is placed in process data in the <diffTimestamp> tags:

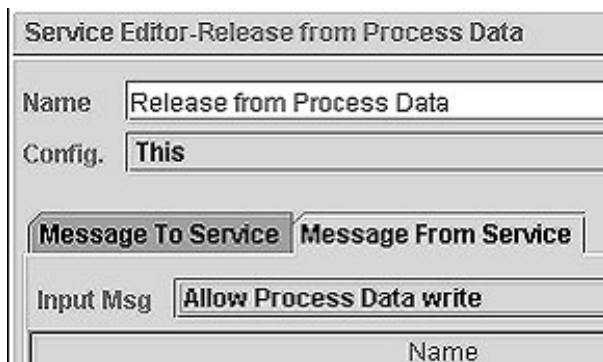
```
<diffTimestamp>
  <elapsedTimeMillis>1543</elapsedTimeMillis>
</diffTimestamp>
```

Release From Process Data (releaseRequest)

This releases the selected node (in this example the primary document) from process data.

Set the following parameter in the GPM Message From Service tab:

Select **Allow Process Data Write** from the Input Msg drop-down list.



Set the following parameters in the GPM Message To Service tab:

1. Select **Messages Only** from the Output Msg drop-down list.

2. Set Message Name to **releaseRequest** and click **Add**.
3. In the Advanced Editor, assign **PrimaryDocument** to **from**, select the Use XPATH? check box, and click **OK**.

Service Editor-Add Document to Process Data

Name: Add Document to Process Data

Config: This

Message To Service: Message From Service

Output Msg: Messages Only

Message Name: takeDocumentRequest

Advanced Editor : Add Document to Process

Name	Value	Use XPATH?
documentId	documentId/text()	<input checked="" type="checkbox"/>
documentKey	'Document1'	<input checked="" type="checkbox"/>

The following illustration shows the BPML:

```
<operation name = "Release from Process Data">
  <participant name="This"/>
  <output message="releaseRequest">
    <assign to="from" from="'PrimaryDocument'"/>
  </output>
  <input message="releaseResponse">
    <assign to="." from="*" />
  </input>
</operation>
```

Process Data – Before Release

Before the This service is called, the process data contains the primary document in the PrimaryDocument node:

Primary document appears in process data.

```
<ProcessData>
  <PrimaryDocument SCIObjectID="L2000-000248:f96c:fae7d336aa:-784a" />
</ProcessData>
```

Process Data – After Release

After the This service has completed, the process data no longer contains the primary document:

```
<ProcessData>  
</ProcessData>
```

Primary document is removed
from process data.

TIBCO Rendezvous Adapter

The following table provides an overview of the TIBCO Rendezvous™ adapter:

System name	Tibco
Graphical Process Modeler (GPM) category	All, Queuing
Description	Connects Application to the TIBCO Rendezvous server, which is the message path for TIBCO ActiveEnterprise™. TIBCO ActiveEnterprise is the TIBCO integration platform that allows applications, services, and third-party systems to be linked together into an integrated system.
Preconfigured?	No
Requires third party files?	Yes. See <i>Required Files</i> on page 1542.
Platform availability	Sun Solaris
Related services	No
Initiates business processes?	Yes
Invocation	Runs as part of a business process.
Restrictions	The TIBCO Rendezvous adapter works only in the Solaris environment.

Requirements

The TIBCO Rendezvous adapter works only in the Solaris environment.

How the TIBCO Rendezvous Adapter Works

The TIBCO adapter supports publish (send) and subscribe (receive) transactions. These transactions are linked using the subject, which is a TIBCO Rendezvous server location where publisher applications can publish messages and subscriber applications can subscribe to messages. Publishers are *decoupled* from subscribers. That is, they do not coordinate data transmission with each other, except by using the subject.

When used as an outbound adapter, the TIBCO adapter publishes transactions to the subject that the TIBCO Rendezvous applications use for subscribe transactions.

When used as an inbound adapter, the TIBCO adapter subscribes to transactions for the subject that the TIBCO Rendezvous applications use for publish transactions.

The TIBCO Rendezvous daemon supports efficient network communications by acting as the pathway for the TIBCO Rendezvous business processes running across the network. The TIBCO Rendezvous daemon arranges the details of data transport, packet ordering, receipt acknowledgment, retransmission requests, and data dispatching for the appropriate business processes.

Data Flow of an Outbound Adapter

Following is the data flow for the TIBCO Rendezvous adapter when it functions as an outbound adapter:

1. The TIBCO adapter outputs data in XML format which is encoded in a `TibrvMsg_XML` wire format. A *wire format* is a universal format independent of hardware, operating system, and programming language architecture that provides a common language to connect diverse programs.
2. After the data is encoded in a `TibrvMsg_XML` wire format, it is sent to the TIBCO Rendezvous server and processed by the TIBCO API.
3. The TIBCO API provides the `TibrvXML` object, which accepts the XML message as a byte array and compresses it, reducing the number of bytes sent over the network.
4. The `TibrvXML` object is stored in the `TibrvMsgField` data field object.
5. The `TibrvMsgField` data field object is added to the `TibrvMsg` object, which is transmitted to the TIBCO Rendezvous server.

Data Flow of an Inbound Adapter

Following is the data flow for the TIBCO Rendezvous adapter when it functions as an inbound adapter:

1. The TIBCO Rendezvous adapter accepts the `TibrvMsg` object. The `TibrvMsg` object contains the `TibrvMsgField`, which contains the compressed `TibrvXML`.
2. The incoming XML data is converted into a byte array and is passed to the business process context, which is passed to Application.

Implementing the TIBCO Rendezvous Adapter

To implement the TIBCO Rendezvous adapter, complete the following tasks:

1. Activate your license for the TIBCO Rendezvous adapter. For information, see *An Overview of Implementing Services*.
2. Create a TIBCO Rendezvous adapter configuration. For information, see *Managing Services and Adapters*.
3. Configuring the TIBCO Rendezvous adapter. For information, see *Configuring the TIBCO Rendezvous Adapter* on page 1541.
4. Use the TIBCO Rendezvous adapter in a business process.

Configuring the TIBCO Rendezvous Adapter

To configure the TIBCO Rendezvous adapter, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Service number	User Datagram Protocol (UDP) service number. The default value is 7500.
Network	IP address of the server running the TIBCO Rendezvous daemon.
Daemon	Name of the background process that completes the information pathway between TIBCO Rendezvous processes across the network. The name has one format, <i>rvd</i> , with <i>rvd</i> indicating the process name.
Action	One of the following modes of EAI communications: <ul style="list-style-type: none"> ◆ async publish (outbound communication) ◆ async subscribe (inbound receiver of events) ◆ sync send (request/response outbound communication)
Subject	TIBCO Rendezvous server location to which the messages are published or subscribed.

Required Files

The following table identifies required files for the TIBCO Rendezvous adapter. These files must be located on the server where the TIBCO Rendezvous applications reside.

Path	Requirement
CLASSPATH	Include the <i>tibrvj.jar</i> file, which contains the TIBCO API class files

Path	Requirement
LD_LIBRARY_PATH	Include the following library (*.so) files (provided by TIBCO): <ul style="list-style-type: none"> ◆ librv.so ◆ librvcm.so ◆ librvcmq.so ◆ librvfst.so ◆ librvft.so ◆ librvjcms11.so ◆ librvjfts11.so ◆ librvjs11.so ◆ librvts.so ◆ librvXt.so ◆ libtibrv.so ◆ libtibrv64.so ◆ libtibrvcm.so ◆ libtibrvcm64.so ◆ libtibrvcmq.so ◆ libtibrvcmq64.so ◆ libtibrvft.so ◆ libtibrvft64.so ◆ libtibrvj.so

TIBCO Rendezvous Publisher Transaction

To publish messages to the TIBCO Rendezvous server, you must configure the adapter to the appropriate subject.

Publisher Configuration

The following table describes the configuration required to run a TIBCO Rendezvous publisher transaction:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Service number	User Datagram Protocol (UDP) service number. The default value is 7500.
Network	IP address of the server running the TIBCO Rendezvous daemon.

Field	Description
Daemon	Name of the background process that completes the information pathway between TIBCO Rendezvous processes across the network. The name has one format, <i>rvd</i> , with <i>rvd</i> indicating the process name.
Action	Of the modes of EAI communications listed, select either: <ul style="list-style-type: none"> ◆ Async publish (outbound communication) ◆ Sync send (request/response outbound communication)
Subject	TIBCO Rendezvous server location to which the messages are published or subscribed.
User Timeout	Field displays when you select sync send from the Action field. This is the time designated for a response to the transmission. Required.

Publisher Transaction Process

Following is the publisher transaction process for the TIBCO Rendezvous adapter:

1. When run as part of a business process, the TIBCO Rendezvous publisher extracts the document from the business process context and converts it into a message.
2. The publisher uses the adapter to publish (send) the message to the specified TIBCO Rendezvous server.
3. If the transmission is synchronous, the adapter waits for a document from the TIBCO Rendezvous server. The document is returned as the primary document. If the transmission is asynchronous, the adapter does not wait for a document but does receive a status indicating whether the transmission was a success or failure.
4. The publisher returns the status of the TIBCO Rendezvous message generation to the business process.

TIBCO Rendezvous Subscriber Transaction

To subscribe to messages from the TIBCO Rendezvous server, you must configure the adapter to the appropriate subject:

Subscriber Configuration

The following table describes the fields used to configure the TIBCO Rendezvous server for a subscriber transaction:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Service number	User Datagram Protocol (UDP) service number. The default value is 7500.
Network	IP address of the server running the TIBCO Rendezvous daemon.
Daemon	Name of the background process that completes the information pathway between TIBCO Rendezvous processes across the network. The name has one format, <i>rvd</i> , with <i>rvd</i> indicating the process name.
Action	Of the modes of EAI communications listed, select async subscribe (inbound receiver of events)
Subject	TIBCO Rendezvous server location to which the messages are published or subscribed.
FileName	File name of the incoming XML document.
Business Process Name	Name of the business process to start.

Subscriber Transaction Process

Following is the subscriber transaction process for an asynchronous transaction:

1. The subscriber subscribes to a particular TIBCO Rendezvous message.
2. When the subscription operation is complete, the TIBCO Rendezvous server sends the subscriber an instance of the message.
3. The subscriber passes the incoming XML document to the initial business process context and runs the specified business process.
4. In the case of an asynchronous subscriber, the subscriber sends an acknowledgment to the TIBCO Rendezvous application that sent the response.
5. The subscriber listens for additional TIBCO Rendezvous messages.

Timestamp Utility Service

The Timestamp Utility service enables you to create, format, and perform arithmetic operations on timestamp values in a business process. The following table provides an overview of the Timestamp Utility service:

System name	TimestampUtilService
Graphical Process Modeler (GPM) category	All Services
Description	Enables you to create, format, and perform arithmetic operations on timestamp values in a business process. The following options are available with the Timestamp Utility service: <ul style="list-style-type: none">◆ Create a timestamp for the current time.◆ Add the numeric values of two timestamps.◆ Subtract one timestamp value from another.◆ Format timestamp values.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Restrictions	No

How the Timestamp Utility Service Works

The Timestamp Utility service enables you to use and manipulate timestamps as part of a business process. One example of how this service could be used would be to calculate the expiration time for incoming documents. If a document had to be acted upon within a two-hour time frame, you could use the service to get the timestamp for when the document arrives, and then calculate the time (two hours later), when an error should be generated if no action has been taken on that document. You could also use the service to generate a timestamp when errors occur in a business process, and use the timestamp as part of the information delivered in an e-mail message about the error, or send the information to a report.

Business Process Examples

The following figure shows process data that uses the Timestamp Utility service to format a timestamp:

```
<operation>
  <participant name="TimestampUtilService"/>
  <output message="xout">
    <assign to="action" from="'format'"/>
    <assign to="baseTime" from="'now'"/>
    <assign to="format">yyyy.MM.dd G 'at' hh:mm:ss z</assign>
  </output>
  <input message="in">
    <assign to="." from="*" />
  </input>
</operation>
```

For examples of timestamp format patterns and results, see <http://java.sun.com> and see the information about the SimpleDateFormat class.

Implementing the Timestamp Utility Service

To implement the Timestamp Utility service, complete the following process:

1. Create a Timestamp Utility service configuration. See *Creating a Service Configuration*.
2. Configure the Timestamp Utility service. See *Configuring the Timestamp Utility Service* on page 1547.
3. Use the Timestamp Utility in a business process.

Configuring the Timestamp Utility Service

The following table describes the fields used to configure the Timestamp Utility service in the GPM:

Field	Description
Config	Name of the service configuration.
action	Action the service should perform. Valid values are current_time, add, diff, and format. Optional.
baseTime	String to use as the base time. Optional. Default time expression is milliseconds; used if no scale is supplied (see scale parameter, below).
offsetTime	Amount to add or subtract from the baseTime. Used with action parameters add and diff only. Optional. Default time expression is milliseconds; used if no scale is supplied (see scale parameter, below).
format	String that describes how to format the parameter. Optional. See http://java.sun.com for information about using SimpleDateFormat values.
scale	How the parameter value is expressed in time scales. Valid values are msec, sec, min, hr, and day. Optional. If not set, the baseTime and offsetTime parameters are expressed in milliseconds.

Timezone Offset Service

The following table provides an overview of the Timezone Offset service, which is used with the Oracle E-Business Suite adapter:

System name	TimezoneService
Graphical Process Modeler (GPM) category	All Services
Description	<p>The Timezone Offset service computes the difference between the local timezone and Greenwich Mean Time (GMT), using the format +/-HHMM. For example, the value for U.S. Eastern time would be -0500 during standard time, or -0400 during daylight savings time.</p> <p>The Timezone Offset service was created for use with the Oracle E-Business Suite adapter.</p>
Business usage	The Open Applications Group (OAG) XML standards use this format for time zone. OAG documents are used by the Oracle E-Business Suite adapter.
Usage example	Use this service to determine the time zone offset value to insert into an OAG document, such as a purchase order or invoice.
Preconfigured?	Yes. An instance named TimezoneService is installed and configured with the Oracle E-Business Suite adapter.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Oracle E-Business Suite adapter (BPML-based adapter)
Application requirements	None
Initiates business processes	No
Invocation	Runs by the Oracle E-Business Suite business processes.
Business process context considerations	None
Returned status values	<ul style="list-style-type: none">◆ Success – The service successfully computed the time zone offset◆ Error – System error
Restrictions	None. Any number of instances may be created, but typically a single instance may be used by any number of business processes.
Persistence level	Default (Full)
Testing considerations	To test it, create a configuration, then run it from a business process as shown in the business process example in <i>Oracle E-Business Suite Adapter</i> .

Implementing the Timezone Offset Service

The Timezone Offset service is implemented as part of the Oracle E-Business Suite adapter configuration; no separate implementation or configuration is required.

Parameters Passed from Service to Business Process

The parameter passed from the Timezone Offset service to the business process is shown in the following table:

Parameter	Description
timezoneOffsetFromGMT	The difference between the local timezone and Greenwich Mean Time (GMT), using the format +/-HHMM. For example, the value for U.S. Eastern time would be -0500 during standard time, or -0400 during daylight savings time

For more information about using the Timezone Offset service, see *Oracle E-Business Suite Adapter*.

Typing Service

The following table provides an overview of the Typing service:

System name	TypingServiceType
Graphical Process Modeler (GPM) categories	All Services, Translation
Description	Replaces specified required and optional parameters into process data. It also can identify the correct map to translate a document into XML.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Invocation	Runs as part of a business process.

How the Typing Service Works

The Typing service uses the first map from a set of maps that:

- Are valid translation objects (end with .txo)
- Have an XML output side
- Are checked in to Application
- Are enabled in Application

The Typing service has a list of required and optional parameters that it searches for in the translated document. If all required parameters are present, the service is complete. If all required parameters are not present, the Typing service uses the next map in the list and translates the primary document. The service uses each map until one of them produces a translation that provides all the required parameters from the primary document. When the service has all of the required parameters, it places the values of the parameters into process data for use by other services.

The following steps summarize how the Typing service works in a business process:

1. The Typing service runs the translator to find pertinent pieces of data from an input file (used for routing and decisions later in the process).
2. The translator loads this data into a Document Object Model (DOM), and the Typing service harness takes the data from the DOM and places it into the business process context.

Note: If the input document character encoding is specified in Application, it overrides the encoding specified in the map. The output document content type and character encoding are set according to the information in the map.

Implementing the Typing Service

To implement the Typing service, complete the following tasks:

1. Activate your license for the Typing service. For information, see *Managing Services and Adapters*.
2. Create a Typing service configuration. For information, see *Managing Services and Adapters*.
3. Configuring the Typing service. For information, see *Configuring the Typing Service*.
4. Use the Typing service in a business process.

Configuring the Typing Service

To configure the Typing service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
edi_input_decimal_separator	Character used to indicate the decimal point on the input side.
edi_input_element_delimiter	Character used to delimit elements (fields) on the input side.
edi_input_release_character	Character used to quote elements (fields) that contain the delimiter on the input side.
edi_input_repeating_element_delimiter	Character used to delimit repeating elements on the input side.
edi_input_segment_delimiter	Character used to delimit segments on the input side.
edi_input_sub_element_delimiter	Character used to delimit subelements on the input side.
edi_input_tag_delimiter	Character used to delimit tags on the input side.
optional_parmlist	Space delimited list of parameters that may be present in the primary document when it is translated by the service. Optional.
required_parmlist	Space-delimited list of parameters that are required to be present in the primary document when it is translated by the service. Generally, these parameters represent the names of XML elements present in the XML output document after a successful translation of the primary document. If these parameters are not all found in the primary document, the Typing service tries the next map in the list (typing maplist) until one of the maps produces a translated document that has all of the required fields. Required.
typing_maplist	Space-delimited list of typing service maps (.txo maps with XML on the output side that have been checked into Application and enabled). Required.

Field	Description
validate_input_against_dtd	Validates the input to the DTD specified in the input document. Valid values are No validation, Validate using a DTD, and Validate using an XML schema.

User Service

The following table provides an overview of the User service:

System name	UserServiceType
Graphical Process Modeler (GPM) categories	All Services, System
Description	Checks the Application database account profiles for permissions that may restrict a user's access to a business process, Web template, or resource.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No

How the User Service Works

The User service is used by both Application and by business processes that reference a User service:

For Application, the User service provides security by verifying permissions set for a user when the user is attempting to run a restricted business process.

A business process uses the User service to extract user data from the Application database for use within the business process as it runs. The user data can be as little as a last name or as much as all the data in the user profile set up by the system administrator. A business process can also reference the User service to help route documents from person to person.

Implementing the User Service

To implement the User service, complete the following tasks:

1. Create a User service configuration. See *Managing Services and Adapters*.
2. Configure the User service. See *Configuring the User Service*.
3. Use the User service in a business process.

Configuring the User Service

To configure the User service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
MethodName	Action that this configuration of the service performs with the database. Valid values are: <ul style="list-style-type: none">◆ getEmail – Obtain the e-mail address associated with the user.◆ getEntityID – Obtain only the entity ID associated with the user.◆ getFirstname – Obtain only the first name of the user.◆ getFullname – Obtain the first and last name of the user.◆ getLastname – Obtain the last name of the user.◆ getPager – Obtain the pager number of the user.◆ getParentID – Obtain the manager's name of the user.◆ getUserToken – Obtain the entire user account profile.◆ hasGroup – Check for groups associated with the user. Requires a value to be typed in the MethodParm field.◆ hasPermission – Check for permissions associated with the user. Requires a value to be typed in the MethodParm field.
MethodParm	Specific permission or group name associated with the user. Required for both hasGroup and hasPermission method names.
UserName	User name that identifies the account profile on which the MethodName parameter conducts the configured action.

Output Results

The following are examples of output results after the User service runs.

Single MethodName return:

```
<UserService>
  <methodName>return value</methodName>
</UserService>
```

User service returning the MethodName *getEmail*:

```
<UserService>
  <getEmail>janedoe@company.com</getEmail>
</UserService>
```

The *getUserToken* method is the only method that returns a node structure within the method. The following is an example of a User service returning the MethodName *getUserToken*, which includes the user token node structure:

```
<UserService>
  <getUserToken>
    <usertoken>
```

```
        <username> </username>
        <firstname> </firstname>
        <lastname></lastname>
        <fullname></fullname>
        <email></email>
        <parentid></parentid>
        <groups>
            <group></group>
        </groups>
        <permissions>
            <perm></perm>
    </permissions>
</usertoken>
    </getUserToken>
</UserService>
```

Vitria (Businessware) Adapter

The following table provides an overview of the Vitria[®] BusinessWare[®] 3.x adapter:

System name	Vitria
Graphical Process Modeler (GPM) category	None
Description	Connects Application to the Vitria [®] BusinessWare [®] Server v3.x, which is the message path for Vitria BusinessWare.
Preconfigured?	No
Requires third party files?	Vitria 3.x. For additional requirements, see <i>Requirements</i> on page 1556.
Platform availability	Sun Solaris
Related services	No
Application requirements	No
Initiates business processes?	Yes
Invocation	Runs as part of a business process.
Restrictions	The Vitria (Businessware) adapter works only in the Solaris environment.

Requirements

The following table identifies files that are required for the Vitria (Businessware) adapter. These files must be located on the server where the Vitria BusinessWare applications reside.

Path	Requirement
CLASSPATH	Include the BW30.jar file, which contains the Vitria API class files.
VTNAME	Environment variable that sets the location of the Vitria BusinessWare server in the network. VTNAME must have the following syntax: [<i>hostname</i> <i>ip-address</i>]: <i>ip-port-number</i> / <i>credential-type</i> example: 127.0.0.1:6000/tcp:

Path	Requirement
LD_LIBRARY_PATH	<p>Include the following library (*.so) files (provided by Vitria):</p> <ul style="list-style-type: none"> ◆ libiconv3_o3gri.so ◆ libnpjavanative3.so ◆ libnpjavanative3_g.so ◆ libnplogger3.so ◆ libnplogger3_g.so ◆ libvFlowBridge.so ◆ libvtlocale3_o3gri.so ◆ libvtlogfil3_o3gri.so ◆ libvtmeta3_o3gri.so ◆ libvtorb3_o3gri.so ◆ libvtORBServLocale3_o3gri.so ◆ libvtTransformerLocale3_o3gri.so ◆ libvtValidatorLocale3_o3gri.so

How the Vitria (Businessware) Adapter Works

The Vitria adapter supports publish (send) and subscribe (receive) transactions. These transactions are linked using the subject, which is a Vitria BusinessWare server location where publisher applications can publish messages and subscriber applications can subscribe to messages. Publishers are *decoupled* from subscribers: that is, they do not coordinate data transmission with each other, except by using the subject.

When used as an outbound adapter, the Vitria adapter publishes transactions to the subject that the Vitria BusinessWare applications use for subscribe transactions.

When used as an inbound adapter, the Vitria adapter subscribes to transactions for the subject that the Vitria BusinessWare applications use for publish transactions.

The Vitria BusinessWare daemon supports efficient network communications by acting as the pathway for the Vitria BusinessWare business processes running across the network. The Vitria BusinessWare daemon arranges the details of data transport, packet ordering, receipt acknowledgment, retransmission requests, and data dispatching for the appropriate business processes.

Data Flow of an Outbound Adapter

Following is the data flow for the Vitria (Businessware) adapter when it functions as an outbound adapter:

1. The Vitria adapter outputs data in XML format which is encoded in an XML wire format. A *wire format* is a universal format independent of hardware, operating system, and programming language architecture that provides a common language to connect diverse programs.
2. After the data is encoded in an XML wire format, it is sent to the Vitria BusinessWare server and processed by the Vitria API.

3. The Vitria API provides the XML object, which accepts the XML message as a byte array and compresses it, reducing the number of bytes sent over the network.
4. The XML object is stored in the message field data field object.
5. The message field data field object is added to the message object, which is transmitted to the Vitria BusinessWare server.

Data Flow of an Inbound Adapter

Following is the data flow for the Vitria (Businessware) adapter when it functions as an inbound adapter:

1. The Vitria (Businessware) adapter continuously receives Vitria BusinessWare objects. The objects contain the message fields, which contain the compressed XML data.
2. The Vitria (Businessware) adapter converts the incoming XML data into a byte array and passes the data to the business process context, which passes the data to Application.
 - a. If the object is a synchronous request, it contains the data and business process that must run in response to the request.
 - b. The adapter spawns a separate worker thread and passes the data to this thread.
 - c. The worker thread runs the started business process, which polls the business process engine until the business process completes.

Implementing the Vitria (Businessware) Adapter

To implement the Vitria (Businessware) adapter, complete the following tasks:

1. Activate your license for the Vitria (Businessware) adapter. See *An Overview of Implementing Services*.
2. Create a Vitria (Businessware) adapter configuration. See *Creating a Service Configuration*.
3. Configure the Vitria (Businessware) adapter. See *Configuring the Vitria (Businessware) Adapter* on page 1558.
4. Use the Vitria (Businessware) adapter in a business process.

Configuring the Vitria (Businessware) Adapter

To configure the Vitria (Businessware) adapter, you must specify settings for the following fields in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	Select one of the options: <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
HostName	IP address of the server running the Vitria BusinessWare applications.
PortNumber	Port number of the server running the Vitria BusinessWare applications.
Event Specification	ID or signature with which an event is registered in the Vitria BusinessWare repository. This information must be provided by the Vitria BusinessWare administrator or business analyst.
Action	One of the following modes of EAI communications: <ul style="list-style-type: none"> ◆ Asynchronous Publish (Outbound) – Outbound communication ◆ Asynchronous Subscribe with BootStrap – Inbound receiver of events ◆ Synchronous Publish (Outbound) – Outbound request/response

Vitria BusinessWare Publisher Transaction

To publish messages to the Vitria BusinessWare server, the adapter must be configured to the appropriate event specification.

Publisher Configuration

The following table describes the configuration required to run a Vitria BusinessWare publisher transaction.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
HostName	IP address of the server running the Vitria BusinessWare applications.
PortNumber	Port number of the server running the Vitria BusinessWare applications.
Event Specification	ID or signature with which an event is registered in the Vitria BusinessWare repository. This information must be provided by the Vitria BusinessWare administrator or business analyst.
Action	Of the modes of EAI communications listed, select either: <ul style="list-style-type: none"> ◆ Asynchronous Publish (Outbound) – Outbound communication ◆ Synchronous Publish (Outbound) – Outbound request/response
TimeOut in milliseconds	Field displays when you select Synchronous Publish from the Action field. This is the time designated for a response to the transmission. Optional.

Publisher Transaction Process

Following is the publisher transaction process for the Vitria (Businessware) adapter:

1. When runs as part of a business process, the Vitria BusinessWare publisher extracts the document from the business process context and converts it into a message.
2. The publisher uses the adapter to publish (send) the message to the specified Vitria BusinessWare server.
3. If the transmission is synchronous, the adapter waits for a returned document from the Vitria BusinessWare server. This document is returned as the primary document. If the transmission is asynchronous, the Vitria adapter does not wait for a document but does receive a status indicating the success or failure of the transmission.
4. The publisher returns the status of the Vitria BusinessWare message generation to the business process.

Vitria BusinessWare Subscriber Transaction

To subscribe to messages from the Vitria BusinessWare server, the adapter must be configured to the appropriate event specification.

Subscriber Configuration

The following table describes the fields used to configure the Vitria BusinessWare server for a subscriber transaction.

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
HostName	IP address of the server running the Vitria BusinessWare applications.
PortNumber	Port number of the server running the Vitria BusinessWare applications.
ChannelPath	File system path for the Vitria channel.
Event Specification	ID or signature with which an event is registered in the Vitria BusinessWare repository. This information must be provided by the Vitria BusinessWare administrator or business analyst.
Action	Of the modes of EAI communications listed, select Asynchronous Subscribe with BootStrap (Inbound receiver of events).
Document Filename	File name of the incoming XML document.

Field	Description
BusinessProcess Poll Interval	Interval at which to poll. Valid value is the number of seconds. Optional.
Business Process	Name of the business process you want to enable or disable. Application displays a list of business processes.

Subscriber Transaction Process

Following is the subscriber transaction process for the Vitria (Businessware) adapter:

1. The subscriber subscribes to a particular Vitria BusinessWare message.
2. When the subscription operation is complete, the Vitria BusinessWare server sends the subscriber an instance of the message.
3. The subscriber passes the incoming XML document to the initial business process context and runs the specified business process.
4. In the case of an asynchronous subscriber, the subscriber sends an acknowledgment to the Vitria BusinessWare application that sent the response.
5. The subscriber listens for additional Vitria Businessware messages.

Wait Notify Service

The following table provides an overview of the Wait Notify service:

System name	WaitNotifyCompleteService
Graphical Process Modeler (GPM) category	None
Description	Provides the ability to join two or more asynchronous subprocesses. It enables you to synchronize on several processes that were originally run asynchronously. The subprocesses do not have to have the same wfd_name.
Business usage	This is an internal only service. It enables another service to wait until a specified set of business processes have completed before going on to the next step of the original process.
Usage example	Used in conjunction with an enhancement to the EDI Develope service to allow a process to wait for all interchanges to be develope before proceeding.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	This service will work with other services that initiate business processes at java coding level.
Application requirements	None
Initiates business processes?	Does not initiate any other business processes, but it does notify the original business process that all pre-selected business processes have completed.
Invocation	Does not run in a business process. It can run only by a service that includes custom java code that triggers this functionality.
Business process context considerations	The original business process has to implement a trigger and writes a record in the workflow_data table where the WaitNotifyCompleteService can check if all selected business processes are done. If so, the service will notify the original business process, which will continue processing. An error in a subprocess will be propagated to the original business process.
Returned status values	Basic: Success – Run without error. Advanced: <ul style="list-style-type: none">◆ Sibling still running – Other bootstrapping subprocess still running, so continue parent business process is not done yet.◆ Continue parent wf id= –This is the last subprocess, and it continued parent business process.

Restrictions The Wait Notify service must be used with a custom service that sets the following fields in the service before starting the subprocesses.

Trigger:

```
iwfc.setOrigWFCID(invoking service wfc)
iwfc.addContentElement("PARENT_SERVICE_NAME"+ invoking
service wfc, wfc.getServiceName)
iwfc.addContentElement("PARENT_WF_ID",bootstrapping wf id)
```

After all subprocesses are started, the service has to call the following:

```
WorkFlowData wfData=new WorkFlowData(Connection);
WorkFlowDataTable wfData = new WorkFlowDataTable(conn);
wfData.setDataString(number of sub wfs kicked off
successfully));
wfData.setDataName(wfc.getWorkFlowContextId());
wfData.setDataKey(wfc.getServiceName() +
wfc.getWorkFlowContextId());
wfData.setWorkFlowInstanceId(wfc.getWorkFlowId());
```

Note: The Wait Notify service is called by the system automatically if the fields in the table, *Output from Business Process to Service*, are set before it calls `InitialWorkFlowContext(iwfc).start()`.

Persistence level	Full
Testing considerations	None

Output from Business Process to Service

The following table describes the fields that must be included in the custom service code to use the Wait Notify service:

Parameter Name (BPML Element Value)	Description
PARENT_SERVICE_NAME (iwfc.addContentElement)	Business process context of the service that starts this service. Required.
PARENT_WF_ID (iwfc.addContentElement)	ID of the business process that starts this service. Required.

Wait Service

The following table provides an overview of the Wait service:

System name	WaitService
Graphical Process Modeler (GPM) categories	All Services, Process Controls
Description	The Wait service allows a business process to wait for a period of time before moving on to the next step in the process. The Wait service does not hold its active thread during this wait time, which can reduce resource consumption.
Business usage	The Wait service allows synchronization of polling events without consuming resources. Use this service to specify a certain wait time after a step in a business process. During the wait time, this service does not use an active thread.
Usage example	A business process with a B2B transmit requires a block of time after a failure before a retry is attempted.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	This service does not initiate a business process. It cannot be used without a business process.
Invocation	Runs as part of a business process.
Business process context considerations	While in its waiting state, this service does not hold an active thread, which frees up system resources.
Returned status values	<ul style="list-style-type: none">◆ Success – Wait time has completed successfully.◆ Error – If wait time value is not found or the value cannot be parsed.
Restrictions	<p>For short wait intervals (less than 30 minutes), Application places the business process in a special "0" queue. From the "0" queue, it regains an active thread when it reaches its expiration, within a range of plus or minus 30 seconds (because the "0" queue checks every 30 seconds for wait times that have completed).</p> <p>For longer wait times, the business process is returned to active by the scheduled system process BPEXpirator. The frequency the BPEXpirator process is scheduled to run (defaults to every 15 minutes) must be inline with intended wait times. The WaitService could potentially wait up to a full BPEXpirator cycle, not being restarted until the next scheduled BPEXpirator execution. For example, if a 45 minute wait expired just after an execution of the BPEXpirator on a 15-minute cycle, that wait will continue until the next BPEXpirator executes, making it a 60 minute, rather than 45 minute wait. By coordinating the wait time and the BPEXpirator schedule, you can configure a window of time that is appropriate for your needs.</p>

How the Wait Service Works

The Wait service performs a function similar to the Sleep service: it allows you to pause a business process for a set length of time. However, unlike the Sleep service, the Wait service does not retain an active processing thread, which frees the thread for other processing jobs. Use the Wait service when you need to put a business process in a Wait state for at least one minute. If you need to put a business process in a wait state for less than a minute, use the Sleep service.

Implementing the Wait Service

Because a configuration of the Wait service is installed with Application, no configuration is necessary.

Output from Business Process to Service

The following table describes the output from the business process to the Wait service:

Parameter	Description
WAIT_INTERVAL	Duration of wait time in minutes.

webMethods Adapter

The following table provides an overview of the webMethods® adapter:

System name	WebMethodsPublisher, WebMethodsSubscriber
Graphical Process Modeler (GPM) category	None
Description	Connects Application to a webMethods® Enterprise server to publish events and to subscribe to events. Communication between the adapter and the server occurs across TCP/IP.
Preconfigured?	No
Required third party files	client50.jar, libawssl50jn.so. See <i>Installing the webMethods Adapter</i> on page 1574.
Platform availability	Sun Solaris
Related services	No
Application requirements	No
Initiates business processes?	Yes
Invocation	Runs as part of a business process.

How the webMethods Adapter Works

The webMethods adapter has two parts and supports four basic operations.

Adapter Components

The webMethods adapter has separate components to publish and subscribe to events:

webMethods Publisher adapter – The webMethods Publisher adapter enables business processes running in Application to publish events to the webMethods Enterprise server. For more information about the webMethods Publisher adapter, see *Implementing the webMethods Publisher Adapter* on page 1575.

webMethods Subscriber adapter – The webMethods Subscriber adapter enables users of Application to subscribe to webMethods events. After receiving an event from the webMethods Enterprise server, the webMethods Subscriber adapter starts a user-specified business process in Application. For more information about the webMethods Subscriber adapter, see *Implementing the webMethods Subscriber Adapter* on page 1578.

Basic Operations

The webMethods adapter supports four basic operations:

Synchronous request/response

Runs a business process asynchronously

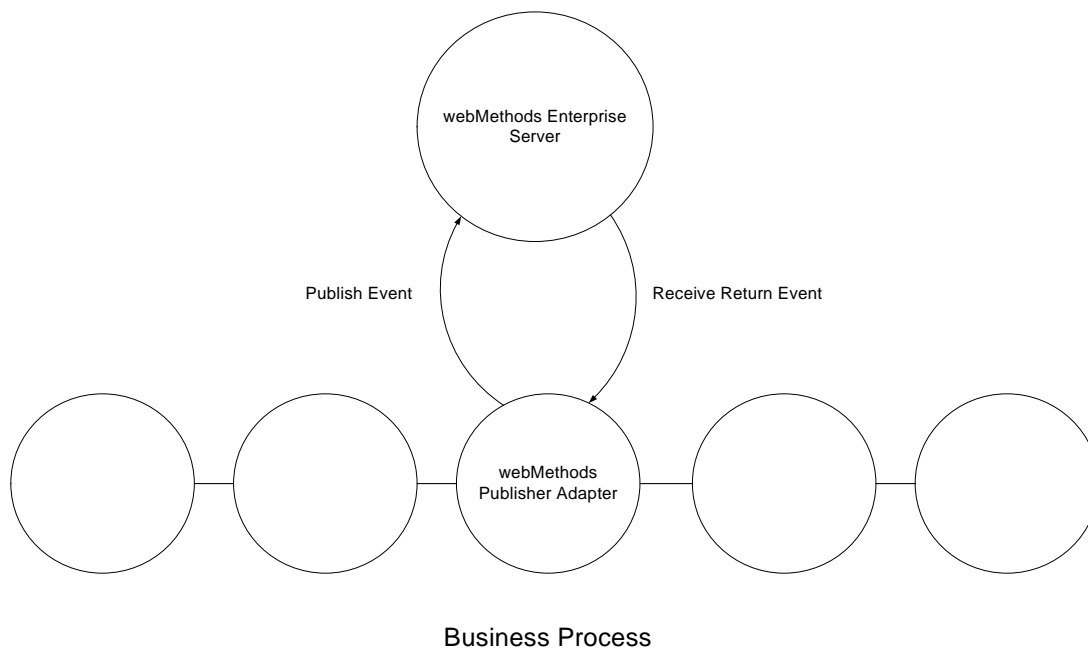
Fire and forget

Asynchronous with transactionality

These operations enable you to perform basic webMethods transactions. The following sections provide details.

Synchronous Request/Response Operation

The following figure shows the synchronous request/response operation for communicating with the webMethods Enterprise server. Only a webMethods Publisher adapter runs in this operation.



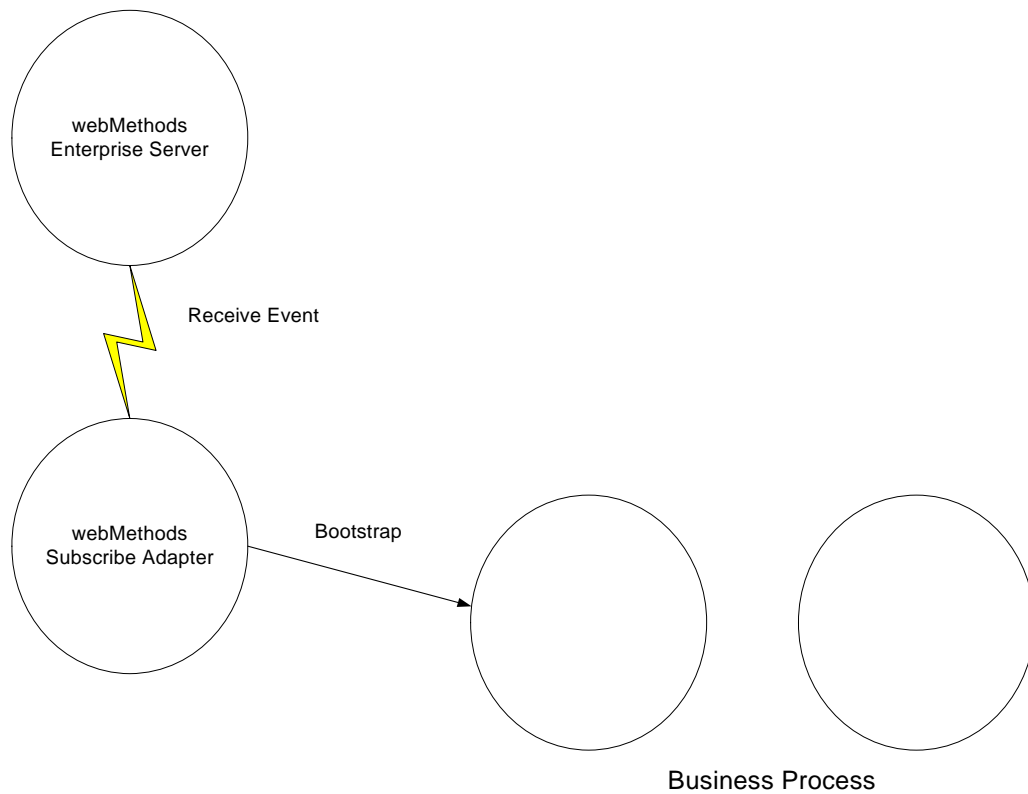
The following steps describe the synchronous request/response operation:

1. The webMethods Publisher adapter translates the primary document from the business process into a webMethods event and publishes the event to the webMethods Enterprise server.
2. The webMethods Publisher adapter waits for a reply event from the webMethods Enterprise server. If the adapter receives a reply event, it translates the webMethods event to an XML document and stores the document as the primary document of the business process. If the adapter does not receive a reply event within the timeout period, it reports the timeout period elapsed before the reply event was received.

Invoke Business Process Asynchronously Operation

The following figure shows starting a business process asynchronously to communicate with the webMethods Enterprise server. Only a webMethods Subscriber adapter runs in this operation.

Note: If successfully configured, a webMethods Subscriber adapter starts automatically when Application starts.

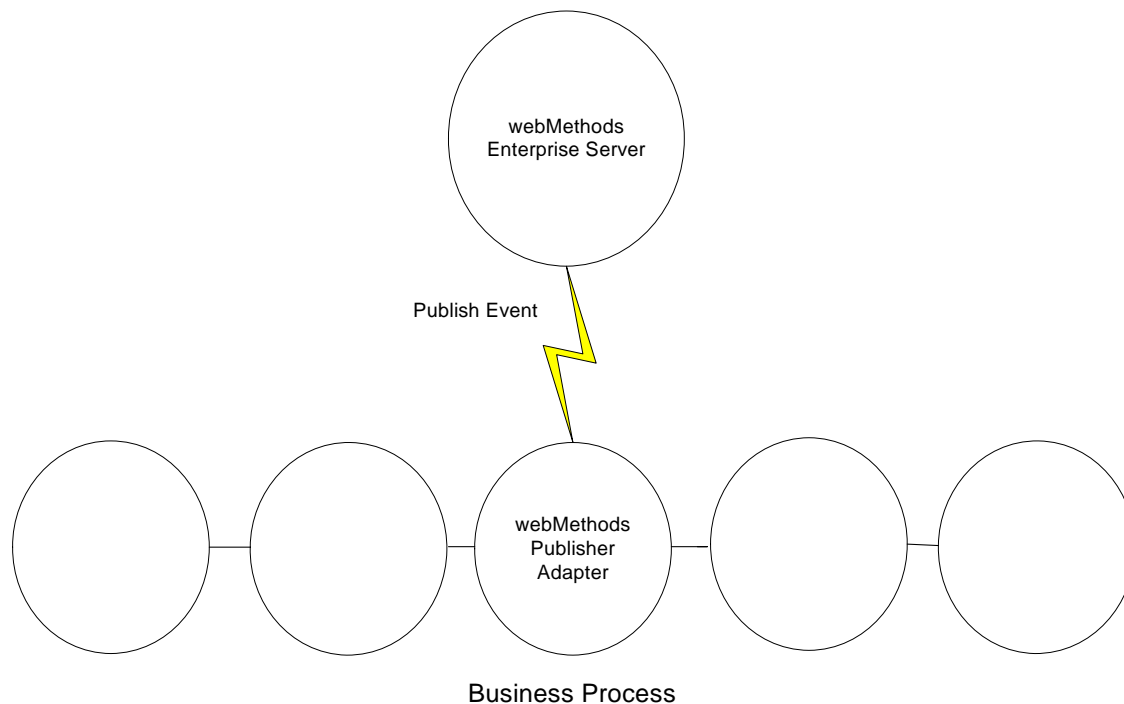


The following steps describe the asynchronous business process start operation:

1. The webMethods Subscriber adapter receives an event to which it has subscribed from the webMethods Enterprise server.
2. The webMethods Subscriber adapter checks its configuration data. If a business process is specified for the received event, the adapter starts that business process. If a business process is not specified for the event, the event is discarded.

Fire-and-Forget Operation

The following figure shows the fire-and-forget operation for communicating with the webMethods Enterprise server. Only a webMethods Publisher adapter runs in this operation.



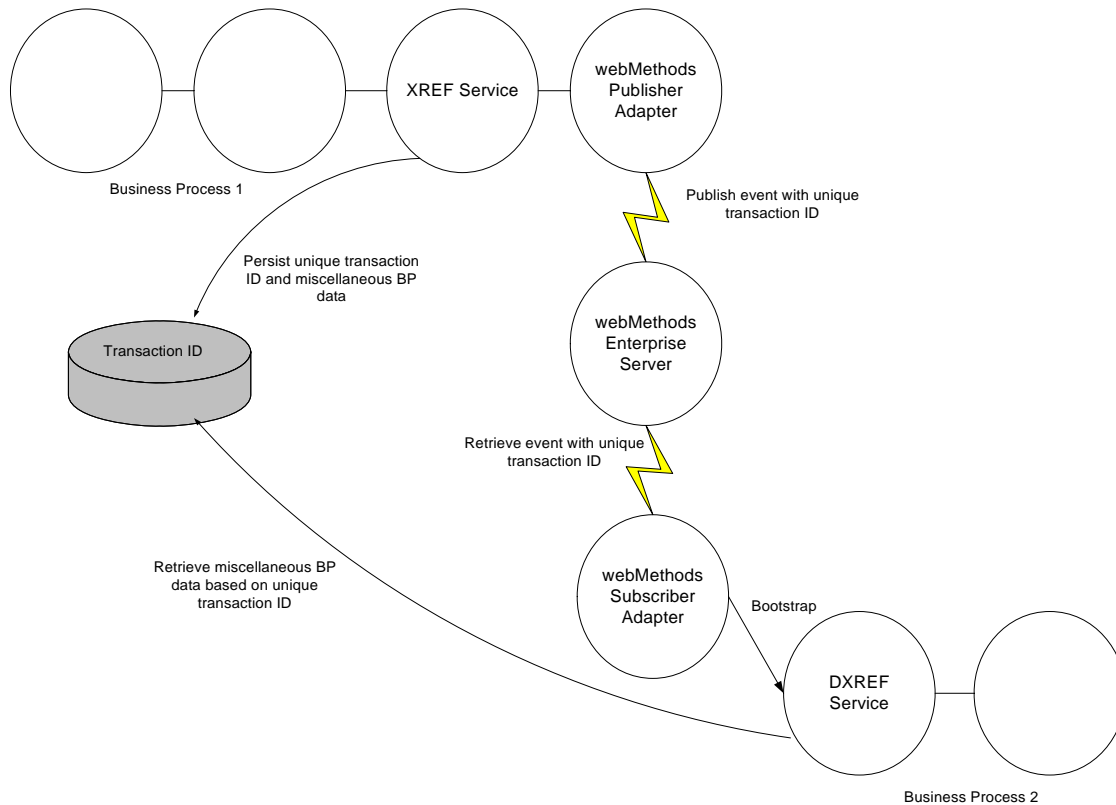
The following steps describe the fire-and-forget operation:

1. The webMethods Publisher adapter translates the primary document from the business process into a webMethods event and publishes the event to the webMethods Enterprise server.
2. The webMethods Publisher adapter waits for an acknowledgment from the webMethods Enterprise server. If the adapter receives an acknowledgment within the timeout period, it reports in the status header of the business process that a subscriber received the event. Otherwise, the adapter reports that no subscriber received the event.

Asynchronous with Transactionality Operation

The asynchronous with transactionality operation runs both a webMethods Publisher adapter and a webMethods Subscriber adapter that connect to the same webMethods Enterprise server. Each transaction is associated with a unique transaction ID.

The following figure shows the asynchronous with transactionality operation:



The asynchronous with transactionality operation has the following steps:

1. The XREF service persists a unique transaction ID as well as miscellaneous business process data in Application.
2. The webMethods Publisher adapter translates the primary document into a webMethods event, puts the transaction ID into the event envelope, and then publishes the event to the webMethods Enterprise server.
3. The webMethods Subscriber adapter receives the event from the webMethods Enterprise server, parses the event body to an XML string, puts the string into the primary document, and gets the transaction ID and puts it in the process data.
4. The DXREF service uses the transaction ID to read the business process data from the Application database and puts it in the process data.

Structure of Input and Output Data

To publish an event to the webMethods Enterprise server, a webMethods Publisher adapter reads the primary document, which is an XML document, from the business process and translates the primary document to a webMethods event. When the webMethods Publisher adapter receives a reply event or the webMethods Subscriber adapter receives an event from the webMethods Enterprise server, the adapter

translates that event to an XML document and sets that document as the primary document of the business process.

The following sections describe the translation between the webMethods event and the XML primary document. Find all the examples in the samples directory.

Basic Structure

The following sample XML document must be translated to an event and published to the webMethods Enterprise server:

```
<WebmethodsAdapter>
  <event name="Sample::GetCustomerAllFields">
    <customerId>14621</customerId>
    <customerFirstName>Joe</customerFirstName>
    <customerLastName>Smith</customerLastName>
  </event>
</WebmethodsAdapter>
```

The root element must always be *WebmethodsAdapter*.

The event information is wrapped in the event element, which has an attribute named *name*. The value of the name attribute is the event type name. The tag name for the event element and its name attribute varies depending on your locale. For more information about locale settings, see *Internationalization* on page 1585.

The fields of the event element are its child elements. The element name is the field name, and the content text is the field value.

The following sample code shows the translation of the XML document to a webMethods event:

```
eventtype Sample::GetCustomerAllFields {
  int customerId;
  unicode_string customerFirstName;
  unicode_string customerLastName;
}
```

Structure Data Field

In a webMethods event, a structure data field is an event field with a user-defined type. It can contain simple data types, arrays, or other structures. When a structure data field is translated to an XML element, the structure name translates as the element tag, and the members of the structure data field translate as the child elements.

The following sample code shows a structure data field in a webMethods event:

```
eventtype Sample::GetCustomerAllFieldsStruct {
  struct {
    int customerId;
    struct {
      unicode_string customerFirstName;
      unicode_string customerLastName;
    } customerName;
  } customer;
  boolean current;
  byte document;
  string socSecNumber;
```

```

    unicode_char billPaid;
}

```

The following sample code shows the translation of the structure data field to XML:

```

<WebmethodsAdapter>
  <event name="Sample::GetCustomerAllFieldsStruct">
    <customer>
      <customerName>
        <customerFirstName>Joe</customerFirstName>
        <customerLastName>Smith</customerLastName>
      </customerName>
      <customerId>14621</customerId>
    </customer>
    <current>true</current>
    <document>0</document>
    <socSecNumber>123-45-6789</socSecNumber>
    <billPaid>Y</billPaid>
  </event>
</WebmethodsAdapter>

```

Sequence Field

When a field of an event is a sequence field, add the attribute *type="seq"* to the element for this field and expand each member of the sequence into an element with the same name as the sequence field. The tag name for the type attribute varies depending on your locale. For more information about locale settings, see *Internationalization* on page 1585.

The following sample code shows a sequence field in a webMethods event:

```

eventtype Sample::GetCustomerAllFieldsSeq {
    string depts[];
}

```

The following sample code shows the translation of the sequence field to XML:

```

<WebmethodsAdapter>
  <event name="Sample::GetCustomerAllFieldsSeq">
    <depts type="seq">
      <depts>sales</depts>
      <depts>marketing</depts>
    </depts>
  </event>
</WebmethodsAdapter>

```

Sequence of Structures Field

When a field of an event is a sequence of structures, the element for this field has two attributes:

One attribute is named *element*, and its value is *struct*.

The other attribute is named *type*, and its value is *seq*.

The tag names for the element and type attributes vary depending on your locale. For more information about locale settings, see *Internationalization* on page 1585.

The following sample code shows a sequence of structures field in a webMethods event:

```

eventtype Sample::GetCustomerAllFieldsSeqOfStructs {

```

```

    struct {
    unicode_string name;
    unicode_string industry;
    } company[];
}

```

The following sample code shows the translation of the sequence of structures field to XML:

```

<WebmethodsAdapter>
  <event name="Sample::GetCustomerAllFieldsSeqOfStructs">
    <company element="struct" type="seq">
      <company>
        <name>ACME</name>
        <industry>food</industry>
      </company>
      <company>
        <name>Computer World</name>
        <industry>computer</industry>
      </company>
      <company>
        <name>Bob's Convenience Store</name>
        <industry>retail</industry>
      </company>
    </event>
</WebmethodsAdapter>

```

Sequence of Sequences Field

When a sequence field in an event is a multidimensional array, the element for that field has two attributes:

One attribute is named *element*, and its value is *seq*.

The other attribute is named *type*, and its value is *seq*.

The tag names for the element and type attributes vary depending on your locale. For more information about locale settings, see *Internationalization* on page 1585.

The following sample code shows a sequence of sequences field in a webMethods event:

```

eventtype Sample::GetCustomerAllFieldsSeqOfSeqs {
  string depts[][][];
}

```

The following sample code shows the translation of the sequence of sequences field to XML:

```

<WebmethodsAdapter>
  <event name="Sample::GetCustomerAllFieldsSeqOfSeqs">

    <!-- sales & marketing depts -->
    <depts element="seq" type="seq">

      <!-- sales -->
      <depts element="seq" type="seq">

        <!-- sales America -->
        <depts type="seq">
          <depts>sales America East</depts>
          <depts>sales America Central</depts>

```

```

<depts>sales America West</depts>
</depts>

<!-- sales Europe -->
<depts type="seq">
<depts>sales Europe East</depts>
<depts>sales Europe Central</depts>
<depts>sales Europe West</depts>
</depts>

</depts>

<!-- Marketing -->
<depts element="seq" type="seq">

<!-- marketing America -->
<depts type="seq">
<depts>mktg America East</depts>
<depts>mktg America Central</depts>
<depts>mktg America West</depts>
</depts>

<!-- marketing Europe -->
<depts type="seq">
<depts>mktg Europe East</depts>
<depts>mktg Europe Central</depts>
<depts>mktg Europe West</depts>
</depts>

</depts>

</depts>

</event>
</WebmethodsAdapter>

```

Installing the webMethods Adapter

To install the webMethods adapter, perform the following steps:

1. Change to the *install_dir/bin* directory.
2. Install the webMethods java API library version 5.0.1 file (*client50.jar*) using the following command:

```
./install3rdParty.sh webmethods version -j absolutePath filename
```

For example:

```
./install3rdParty.sh webmethods 5.0 -j /home/webmethods/client50.jar
```

3. If you want the webMethods adapter to support secured socket layer (SSL), install the webMethods SSL shared library file (*libawssl50jn.so*) using the following command:

```
./install3rdParty.sh webmethods -1 /
```

For example:

```
./install3rdParty.sh webmethods 5.0 -1 /home/webmethods/libawssl50jn.so
```

Implementing the webMethods Publisher Adapter

The webMethods Publisher adapter enables business processes that run in Application to publish events to a webMethods Enterprise server.

You can create multiple configurations of the webMethods Publisher adapter. Because the client group defines which events a webMethods Publisher adapter has permission to publish, you must configure a separate instance of the adapter for each client group.

To implement the webMethods Publisher adapter for use in a business process:

1. Create webMethods Publisher adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the webMethods Publisher adapter. For information, see *Configuring the webMethods Publisher Adapter* on page 1575.
3. Use the webMethods Publisher adapter in a business process.

Configuring the webMethods Publisher Adapter

Application Configuration

The following table describes the fields used to configure a webMethods Publisher adapter in Application:

Note: The names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .
Webmethods Host (webmethodsHost)	Host name of the webMethods Enterprise server with which the adapter communicates. Required.
Webmethods Port (webmethodsPort)	Port number of the webMethods Enterprise server with which the adapter communicates. Required.
Broker Name (webmethodsBrokerName)	Name of the webMethods broker client with which the adapter communicates. Required.

Field	Description
Client Group (webmethodsClientGroup)	Name of the client group, which defines the events the adapter has permission to generate. The client group must exist on the webMethods Enterprise server. Required.
Certification File (certFile)	Absolute path to the certificate used for secure authentication and SSL encryption. Optional.
Password (certPassword)	Password for the certificate used for secure authentication and SSL encryption. Optional.
Distinguished Name (certDN)	Distinguished name for the certificate used for secure authentication and SSL encryption. Optional.

GPM Configuration

The following table describes the fields used to configure a webMethods Publisher adapter in the GPM:

Field	Description
Config	Name of the adapter configuration.
webmethodsEventName	Fully qualified event name (for example, Sample::GetCustomer). Optional.
webmethodsSynchronousFlag	Mode of communication with the webMethods Enterprise server: <ul style="list-style-type: none"> ◆ true is synchronous. ◆ false is asynchronous. Required.
webmethodsTimeOut	Amount of time in milliseconds to wait for an acknowledgment or a return event. Required.
language	Language used for internationalization. If no value is specified, English is the default. Optional.
country	Country used for internationalization. Optional.
variant	Vendor- or browser-specific value used for internationalization. Optional.

Sample BPML

The following sample BPML is for a webMethods Publisher adapter:

```

...
<operation name="Webmethods Publisher">
  <participant name="Webmethods Publisher" />
  <output message="WebmethodsInputMessage">
    <assign to="." from="*" />
    <assign to="webmethodsTimeOut">5000</assign>
    <assign to="webmethodsSynchronousFlag">true</assign>
    <assign to="language">fr</assign>
    <assign to="country">FR</assign>
  </output>
  <input message="inmsg">

```



```
<assign to="." from="*"></assign>
</input>
</operation>
...
```

This sample BPML:

Specifies `webmethodsTimeout` and `webmethodsSynchronousFlag`, because they are required. If they are not specified, you receive an error when the business process runs.

Does not specify `webmethodsEventName` because it is not required by the `webMethods` adapter.

Does not specify `webmethodsHost`, `webmethodsPort`, `webmethodsBrokerName`, and `webmethodsClientGroup`. The `webMethods` adapter assumes these parameters are configured in `Application`. If they are not specified in `Application`, you receive an error when the business process runs.

Does not specify `certFile`, `certPassword`, and `certDN`, because they are not required by the `webMethods` adapter. They may or may not be configured in `Application`. If they are not specified in `Application`, no security is used.

Specifies language and country, in this case, French as the language and France as the country. The adapter assumes the tags in the incoming document are in French.

Find this sample BPML in the samples directory.

Data Flow

The following description of the process to publish an event with the `webMethods Publisher` adapter assumes the adapter is already configured in `Application`:

1. When a business process needs to publish an event to the `webMethods Enterprise` server, it starts a `webMethods Publisher` adapter.
2. The `webMethods Publisher` adapter converts the incoming primary document to a `webMethods` event and publishes the event to the `webMethods Enterprise` server.
3. If the configured mode of communication is synchronous, an output document contains the information returned to the `webMethods Publisher` adapter from the `webMethods Enterprise` server. If the configured mode of communication is asynchronous, no output document is returned.
4. When the event is published successfully, the `webMethods Publisher` adapter returns a status of `Success` to `Application`.
5. `Application` performs the next step in the business process.

Error Messages

When a business process runs a webMethods Publisher adapter, the adapter can return the following error messages in the Advance Status field. The adapter logs more detailed error information in the log files.

Error Message	Description
Error occurred when creating a broker client	One of the following configuration parameters was invalid: <ul style="list-style-type: none">◆ Webmethods Host◆ Webmethods Port◆ Broker Name◆ Client Group
Broker can not publish this event	Specified event was not in the CanPublish list for the client group.
Error happened when parsing the incoming XML document	Either the XML document was not well-formed, or the structure of the XML document did not match the event to be published.
Timed out before the reply event is received	Adapter, communicating in synchronous mode with the webMethods Enterprise server, did not receive a reply event within the specified timeout period.
Received unexpected acknowledgment when publishing synchronously	Adapter, communicating in synchronous mode with the webMethods Enterprise server, expected a reply event but received an acknowledgment instead.
Error happened while invoking this adapter	Error occurred when a business process started the adapter. This is a general-purpose error message. For more details, check the log file.

Implementing the webMethods Subscriber Adapter

The webMethods Subscriber adapter enables Application users to subscribe to webMethods events. When an event is received from the webMethods Enterprise server, a user-specified business process is started in Application.

When you create and save a configuration of the webMethods Subscriber adapter, the adapter starts and subscribes to all events specified in Application. No business process can explicitly run this adapter. You must specify a business process related to each event. This is the business process Application starts when it receives the specified event from the webMethods Enterprise server.

To implement the webMethods Subscriber adapter for use in a business process:

1. Create a webMethods Subscriber adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the webMethods Subscriber adapter. For information, see *Creating a webMethods Subscriber Adapter Configuration* on page 1579.
3. Use the webMethod Subscriber adapter in a business process.

Creating a webMethods Subscriber Adapter Configuration

The following tables describe the fields used to configure a webMethods Subscriber adapter in Application and connections for the webMethods Subscriber adapter in a property file.

Application Configuration

The following table describes the fields used to configure the webMethods Subscriber adapter in Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .
Webmethods Host	Host name of the webMethods Enterprise server with which the adapter communicates. Required.
Webmethods Port	Port number of the webMethods Enterprise server with which the adapter communicates. Required.
Broker Name	Name of the webMethods broker client with which the adapter communicates. Required.
Client Group	Name of the client group, which defines the events the adapter has permission to generate. The client group must exist on the webMethods Enterprise server. Required.
Certification File	Absolute path to the certificate used for secure authentication and SSL encryption. Optional.
Password	Password for the certificate used for secure authentication and SSL encryption. Optional.
Distinguished Name	Distinguished name for the certificate used for secure authentication and SSL encryption. Optional.

Field	Description
Event x Business Process x	<p>Where x is a number from 1 to 20. Specify up to 20 pairs of events and business process combinations. The event is the fully qualified event name (for example, Sample::GetCustomer) or a wildcard (*). The business process is the name of the business process to start upon receipt of the associated webMethods event. At least one event and business process combination is required; additional event and business process pairs are optional.</p> <p>The following rules apply to the pairs of events and business processes:</p> <ul style="list-style-type: none"> ◆ Each entry contains only one event name (or a wildcard) and only one business process name. ◆ Each event can have only one entry. ◆ Multiple events can be associated with the same business process. Each association requires a separate entry. ◆ A wildcard in the event field associates all remaining events for the client group to the specified business process. ◆ A wildcard is not supported in the business process field. ◆ If no business process is associated with an event, an error occurs when the adapter configuration is saved in Application. ◆ If more than one wildcard is specified for an event, an error occurs when the adapter configuration is saved in Application.

To create a webMethods Subscriber adapter with different hosts, ports, broker names, client groups, or security parameters, you must create multiple configurations of the adapter. Each configuration defines a unique set of parameters.

Configuring the Property File

The webMethods Subscriber adapter polls the webMethods Enterprise server periodically to determine whether the adapter is still connected to the server. You must configure connection management parameters in the property file. The webmethods.properties file is located in *install_dir*/properties/adapter_conf, where *install_dir* is your Application installation directory.

The following table describes the fields used to configure the connection management parameters in the property file:

Field	Description
Connection_Polling_Interval	<p>How often the connection between the adapter and the webMethods Enterprise server is checked.</p> <p>You can specify a number with the following suffixes:</p> <ul style="list-style-type: none">◆ S or s for seconds◆ M or m for minutes◆ H or h for hours◆ D or d for days <p>If no suffix is specified, minutes is the default. The default value is one minute (1M).</p>
Connection_Polling_Interval_Disconnected	<p>How often the connection between the adapter and the webMethods Enterprise server is checked after the connection is down.</p> <p>You can specify a number with the following suffixes:</p> <ul style="list-style-type: none">◆ S or s for seconds◆ M or m for minutes◆ H or h for hours◆ D or d for days <p>If no suffix is specified, minutes is the default. The default value is 10 minutes (10M).</p>
Disconnected_Timeout_Period	<p>How long a downed connection between the adapter and the webMethods Enterprise server is checked before the adapter is shut down.</p> <p>You can specify a number with the following suffixes:</p> <ul style="list-style-type: none">◆ S or s for seconds◆ M or m for minutes◆ H or h for hours◆ D or d for days <p>If no suffix is specified, days is the default. The default value is one day (1D).</p>

Data Flow

The webMethods Subscriber adapter follows this process to subscribe to events:

1. The user specifies all the required parameters in Application as well as all the events to subscribe to and the business processes to start, and then saves the adapter configuration.
2. The webMethods Subscriber adapter validates the configuration parameters. If the parameters are valid, the webMethods Subscriber adapter starts and subscribes to all the events in the webMethods Enterprise server that were specified in Application.

If any of the parameters are invalid, an error is logged.

3. The webMethods Enterprise server publishes the events the webMethods Subscriber adapter subscribes to, and the adapter starts the associated business processes in Application.

The adapter runs until one of the following events occurs:

You stop the webMethods Subscriber adapter from Application.

You update and save the adapter configuration in Application, and it contains invalid parameters.

The connection to the webMethods Enterprise server goes down and is not reestablished.

When you update the configuration of a webMethods Subscriber adapter—change the Webmethods Host, Webmethods Port, Broker Name, Client Group, Certification File, Password, or Distinguished Name parameter—the connection to the webMethods Enterprise server ends, and a new connection is made.

Error Messages

The webMethods Subscriber adapter starts when you save its configuration, either new or updated, in Application. If the adapter starts successfully, no error logs are created. If the adapter does not start successfully, errors are logged and the adapter does not run.

The webMethods Subscriber adapter can log the following error messages:

Error Message	Description
Exception happened when creating broker client	One of the following configuration parameters was invalid: <ul style="list-style-type: none"> ◆ Webmethods Host ◆ Webmethods Port ◆ Broker Name ◆ Client Group ◆ Certification File ◆ Password ◆ Distinguished Name
Error happened when starting connection manager	The connection manager, which controls the connection between the adapter and the webMethods Enterprise server, did not start correctly.
Need to have a BP for event <i>eventName</i>	The specified event (<i>eventName</i>) did not have an associated business process.
Cannot find event with name <i>eventName</i>	The client group did not include the specified event (<i>eventName</i>).
Already defined one default BP for <i>businessProcess</i>	A wildcard (*) was used as an event for more than one business process (<i>businessProcess</i>).
Already defined one BP <i>businessProcess</i> for event <i>eventName</i>	The specified event (<i>eventName</i>) was entered more than once with different business process names.
Need to subscribe to at least one event	No event was specified in the configuration in Application.

Error Message	Description
Error happened when subscribing to event <i>eventName</i>	An error occurred when the adapter tried to subscribe to an event (<i>eventName</i>) in the webMethods Enterprise server.
Error happened when unsubscribing to event <i>eventName</i>	An error occurred when the adapter tried to unsubscribe to an event (<i>eventName</i>) in the webMethods Enterprise server.
Error happened when registering callback	The adapter could not register the callback.
Error happened when listening for subscriptions	The broker client could not listen for subscriptions.
Error happened when waiting for ongoing BPs	When the adapter configuration is updated or when the connection between the adapter and the webMethods Enterprise server goes down, a new broker client and subscriptions are created. If something goes wrong during this process, the adapter checks if it is processing any events from the webMethods Enterprise server. If events are being processed, the adapter waits until the processing completes before it stops. This error occurred when something went wrong before the events completed processing.
Error disconnecting client from webMethods	The adapter needed to stop, but could not disconnect correctly from the webMethods Enterprise server.
Error happened when retrieving valid events from webMethods	An error occurred when the adapter tried to retrieve all the events that the client group could subscribe to from the WebMethods Enterprise server.

DTD Generator

The webMethods DTD generator is a tool that enables you to:

- Create a DTD to validate incoming XML for a webMethods event.

- Retrieve a list of events from webMethods for a client group.

Use the DTD generator from a computer that has an instance of Application with the DTD generator and can communicate with the webMethods Enterprise server.

To use the DTD generator, run the following command under your `${INSTALL_DIR}/bin` directory:

```
run WMDtdGenerator.sh
```

The DTD generator provides both a command line interface and an interactive interface. Use the following syntax for the command line interface:

```
run WMDtdGenerator.sh -h <host> -p <port> -c <client group> -b <broker name> -e <event name> -f <output file name> -l -noDTD
```

The following table describes the options you can define on the command line:

Option	Definition
-h <host>	Host name of the webMethods Enterprise server.
-p <port>	Port number of the webMethods Enterprise server.

Option	Definition
-c <client group>	Name of the webMethods-defined client group.
-b <broker name>	Name of the broker on the webMethods Enterprise server.
-e <event name>	Fully qualified event name (for example, Sample::GetCustomer).
-f <output file name>	Path of the file where you want to write the DTD. To write the DTD to the console, type none as the output file name. If the DTD generator cannot create the output file, it writes the DTD to a file in the default temporary file directory and indicates the location of the file in a message to the console.
-l	Option to retrieve a list of events for the specified client group. Do not include the -e or -f option. Do include the -noDTD option; otherwise, the DTD generator assumes you want to generate a DTD and prompts you for the information needed to generate that DTD.
-noDTD	Use with the -l option to indicate you do not want to generate a DTD.

The DTD generator prompts for any information it needs that you do not provide on the command line. To run the DTD generator in interactive mode, do not include any options and it will prompt you for them.

Note: To pass a value that contains spaces, enclose the value in double quotes, for example, `runWMDtdGenerator.sh -h 10.10.10.10 -p 9999 -b "Broker #1"`

Security

The webMethods software provides support for authentication using digital certificates. The webMethods adapter supports communicating with the webMethods Enterprise server through SSL and certificates. Authentication is optional depending on how you configure the webMethods adapter and the webMethods Enterprise server to meet user security requirements. For information about SSL support of the webMethods Enterprise server as well as how to obtain a certificate, see the webMethods documentation.

The webMethods adapter supports the following security options:

Server-side authentication with encryption – You must configure the certificate file and password. The default distinguished name is used.

Server and client-side authentication with encryption – You must configure the certificate file, password, and distinguished name.

No authentication or encryption – If the client group that owns the adapter does not require authentication, no configuration is required.

For information about configuring the security parameters (certificate file, password, distinguished name), see *Configuring the webMethods Publisher Adapter* on page 1575 and *Creating a webMethods Subscriber Adapter Configuration* on page 1579.

For the webMethods adapter to use SSL to communicate with the webMethods Enterprise server, you must also specify the SSL shared library file. For information about installing the SSL shared library, see *Installing the webMethods Adapter* on page 1574.

Internationalization

You can specify language, country, and variant in the GPM. These parameters define the internationalization rules the webMethods Publisher adapter uses when translating the input and output documents. If you do not specify language, country, or variant, the adapter uses English.

The following tags in the input and output documents are affected by internationalization:

- event
- name
- element
- type

For information about how the adapter translates input and output documents, see *Structure of Input and Output Data* on page 1570.

WebSphere MQ Adapter

The WebSphere MQ adapter communicates with WebSphere MQ to send and receive messages. The WebSphere MQ adapter is most useful for simple communications done in one session. Application also provides the WebSphereMQ Suite, which is a set of services that provide maximum flexibility and functionality by enabling you to script a complete MQ session using a business process. The suite also includes a new adapter that you can use for asynchronous receiving, the WebSphereMQ Async Receiver adapter.

The following table provides an overview of the WebSphere® MQ adapter:

System name	WebSphereMQ
Graphical Process Modeler (GPM) categories	All Services, Messaging
Description	Communicates with WebSphere MQ to send and receive messages.
Business usage	Used to communicate messages with new or existing business applications across different platforms using WebSphere MQ.
Usage example	Inventory information from multiple distribution points is sent to WebSphere MQ where the messages are then retrieved by the WebSphere MQ adapter and processed.
Preconfigured?	No
Requires third party files?	<p>The jar files to be installed as third party files vary with the different versions of WebSphere MQ. You can view the list of jar files from <i>install_dir</i>/java/lib on the WebSphere MQ server. For MQ version 7.0.x and above, all jar files contained in the <i>install_dir</i>/java/lib directory should be installed using the Application install3rdParty utility.</p> <p>Note: The WebSphere MQ adapter may work with previous versions of the WebSphere MQ classes for Java; however it was designed for use with the version 5.3.x or higher. In all cases, the version of the WebSphere MQ jar files installed on the Application must match the version running on the WebSphere MQ server.</p>
Platform availability	All supported Application platforms
Related services	No
Application requirements	WebSphere MQ server
Initiates business processes?	This adapter can be used to initiate business processes once data is received.
Invocation	Used in a business process only to send data or receive data synchronously. An adapter instance that is configured to receive asynchronously cannot be used in a business process. This is because as soon as the adapter is configured to receive asynchronously, it is already running. Trying to execute it would generate an error message.
Business process context considerations	When sending data, the adapter first looks for the ProcessData variable MQ/documentId. If this variable exists, the adapter attempts to retrieve this document. Otherwise it uses the primary document. When receiving data, a document is created in ProcessData and the ID of the document is stored in the variable MQ/documentId.

Returned status values	<ul style="list-style-type: none"> ◆ Success – Successful completion ◆ Error – Some error occurred - check the advanced status and/or system logs
Restrictions	This is a stateful adapter, which means it maintains the state of certain variables for the life of the adapter, such as connections. When configuring an asynchronous receive instance of this adapter, once the configuration is saved and activated, it automatically begins to do a continuous receive until the adapter is shutdown.
Testing considerations	Debug messages can be turned on as part of the adapter configuration without having to turn on full system log debugging. MQ tracing can also be activated as part of the adapter configuration.

How the WebSphere MQ Adapter Works

The WebSphere MQ adapter exchanges messages with a WebSphere MQ server by means of the IBM WebSphere MQ classes for Java API (com.ibm.mq.jar). This API gives the WebSphere MQ adapter the full range of features available for exchanging messages appropriate for your needs. For best performance, the adapter is stateful so that it can connect and maintain the connections for the life of the adapter even when no messages are being exchanged with the server.

Business Process Example

The following example business process illustrates performing a synchronous receive with a correlation identifier match option from the previous send. Since the default action of the send is to copy the message identifier to the correlation identifier, we use the MQ/sentMsgId variable from the send and set the correlation identifier match option so that it only receives messages relevant to the message just sent – which in this case should be the confirmation-on-arrival report.

```
<process name = "MQSndRcv">
  <sequence>
    <!-- This operation sends data to MQ and requests confirmation-on-arrival reports
to be generated with partial data -->
    <operation name="WebSphere MQ Adapter">
      <participant name="MqSnd"/>
      <output message="WebsphereMQInputMessage">
        <assign to="." from="*"></assign>
        <assign to="snd_MQMD_msgType">DATAGRAM</assign>
        <assign to="snd_MQRO_coa">WITH_DATA</assign>
        <assign to="snd_MQMD_replyToQ">Queue_reports</assign>
        <assign to="snd_MQMD_replyToQM">QueueManager</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
```

```
<!-- This operation performs a synchronous receive with a correlation identifier match
option from the previous send. Since the default action of the send is to copy the
message identifier to the correlation identifier, we use the MQ/sentMsgId variable
from the send and set the correlation identifier match option so that it only
receives messages relevant to the message just sent - which in this case should be the
```

```

confirmation-on-arrival report. Since MQ/sentMsgId is a hex string,
rcv_HexToByte_corId is used to convert the hex back to a byte array. -->
  <operation name="WebSphere MQ Adapter">
    <participant name="MqRcvSync" />
    <output message="WebsphereMQInputMessage">
      <assign to="." from="*"></assign>
      <assign to="rcv_MQGMO_wait">Yes</assign>
      <assign to="rcv_MQGMO_waitInterval">10000</assign>
      <assign to="rcv_MQMO_corId" from="MQ/sentMsgId" />
      <assign to="rcv_HexToByte_corId">Yes</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
</process>

```

Implementing the WebSphere MQ Adapter

To implement the WebSphere MQ adapter, complete the following tasks:

1. Install the WebSphere MQ adapter. For information, see *Installing the WebSphere MQ Adapter*.
2. Create a WebSphere MQ adapter configuration. For information, see *Managing Services and Adapters*.
3. Configure the WebSphere MQ adapter. For information, see *Configuring the WebSphere MQ Adapter*.
4. Use the WebSphere MQ adapter service in a business process.

Installing the WebSphere MQ Adapter

Before you can install the WebSphere MQ adapter, you must obtain a vendor library package. The WebSphere MQ Classes for Java (com.ibm.mq.jar) file must be installed on the host computer where Application is installed.

Note: For more information about the WebSphere MQ Classes for Java, go to the IBM WebSphere MQ site at <http://www-306.ibm.com/software/integration/wmq/>.

To install the WebSphere MQ adapter (6.x or lower version):

1. From the computer where the WebSphere MQ server is installed, copy the com.ibm.mq.jar file to a directory on your local server.
2. From the directory on your local server, copy the com.ibm.mq.jar file to a directory on the host computer where Application is installed. In the space below, record the path and name of the directory.
Path: _____
3. Shut down Application if it is running.
4. From the bin directory where Application is installed, install the vendor library package (com.ibm.mq.jar) by running the install3rdParty script included with Application. Use the WebSphere

MQ server version number and the path and name of the directory that you documented in step 2 to enter commands similar to the following examples:

Note: The exact names and version numbers of the vendor's filenames listed in the following examples may vary.

- ◆ On a Unix system, enter:

```
install3rdParty.sh ibm 5_3 -j directory/com.ibm.mq.jar
```

- ◆ On a Windows system, enter:

```
install3rdParty.cmd ibm 5_3 -j directory/com.ibm.mq.jar
```

5. Restart Application.

To install the WebSphere MQ adapter (7.0.x or higher version):

1. From the computer where the WebSphere MQ server is installed, copy all the jar files in the *install_dir/java/lib* directory to a directory on your local server. No files located in the subdirectories under the WebSphere *install_dir/java/lib* directory need to be copied to the directory on your local server.
2. From the directory on your local server, copy the jar files to a directory on the host computer where Application is installed. In the space below, record the path and name of the directory.

Path: _____

3. Shut down Application if it is running.
4. From the bin directory where Application is installed, install all the vendor library packages by running the *install3rdParty* script included with Application. Use the WebSphere MQ server version number and the path and name of the directory that you documented in step 2 to enter commands similar to the following examples:

Note: The exact names and version numbers of the vendor's filenames listed in the following examples may vary.

- ◆ On a Unix system, enter:

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.commonservices.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.defaultconfig.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.fta.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.headers.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.jmqi.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.jms.Nojndi.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.pcf.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.postcard.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.soap.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mq.tools.ras.jar
```

```
install3rdParty.sh ibm 7_0 -j directory/com.ibm.mqetclient.jar
```

- ◆ On a Windows system, enter:

```
install3install3rdParty.cmd ibm 7_0 -j
directory/com.ibm.mq.commonservices.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.defaultconfig.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.fta.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.headers.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.jmqi.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.jms.Nojndi.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.pcf.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.postcard.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.soap.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mq.tools.ras.jar
install3rdParty.cmd ibm 7_0 -j directory/com.ibm.mqetclient.jar
```

5. Restart Application.

Configuring the WebSphere MQ Adapter

To configure the WebSphere MQ adapter, you must specify field settings in Application and in the Graphical Process Modeler (GPM). For general information about service and adapter configurations, see *Managing Services and Adapters*.

Application Configuration

The following table describes the fields used to configure the WebSphere MQ adapter in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Host Name (mq_hostname)	The host name or IP address of the WebSphere MQ server. Optional. If specified, a “client mode” connection will be made. If left blank, a “bindings mode” connection will be made.
Listening Port (mq_port)	The listening port of the WebSphere MQ server. Optional if Host Name is blank (“bindings mode” connection), required otherwise (“client mode” connection). Valid value is any valid numeric port number.
Queue Manager (mq_qManager)	The Queue Manager name to use. Optional. If left blank, the default Queue Manager is used. Alphanumeric.
Queue Name (mq_qName)	The Queue Name to use. Required. Alphanumeric.
Server Connection Channel (mq_svrConxChan)	The Server Connection Channel to use. Optional if Host Name is blank (“bindings mode” connection), required otherwise. Alphanumeric.
User ID (mq_userId)	A user identifier, if required, to access the WebSphere MQ server. Optional. Alphanumeric.
Password (mq_password)	A user password, if required, to access the WebSphere MQ server. Optional. Alphanumeric.

Field	Description																																						
CCSID (mq_ccsid)	<p>If needed, enter the Coded Character Set Identifier (CCSID) that represents the codeset name you wish to use. Optional.</p> <table border="1"> <thead> <tr> <th>Codeset name</th> <th>Use CCSID</th> </tr> </thead> <tbody> <tr><td>ISO 8859-1</td><td>819 (default)</td></tr> <tr><td>ISO 8859-2</td><td>912</td></tr> <tr><td>ISO 8859-3</td><td>913</td></tr> <tr><td>ISO 8859-5</td><td>915</td></tr> <tr><td>ISO 8859-6</td><td>1089</td></tr> <tr><td>ISO 8859-7</td><td>813</td></tr> <tr><td>ISO 8859-8</td><td>916</td></tr> <tr><td>ISO 8859-9</td><td>920</td></tr> <tr><td>ISO 8859-13</td><td>921</td></tr> <tr><td>ISO 8859-15</td><td>923</td></tr> <tr><td>big5</td><td>950</td></tr> <tr><td>eucJP</td><td>954, 5050, 33722</td></tr> <tr><td>eucKR</td><td>970</td></tr> <tr><td>eucTW</td><td>964</td></tr> <tr><td>eucCN</td><td>1383</td></tr> <tr><td>PCK</td><td>943</td></tr> <tr><td>GBK</td><td>1386</td></tr> <tr><td>koi8-r</td><td>878</td></tr> </tbody> </table> <p>Note: CCSID is not used when connecting directly using “binding moce.”</p>	Codeset name	Use CCSID	ISO 8859-1	819 (default)	ISO 8859-2	912	ISO 8859-3	913	ISO 8859-5	915	ISO 8859-6	1089	ISO 8859-7	813	ISO 8859-8	916	ISO 8859-9	920	ISO 8859-13	921	ISO 8859-15	923	big5	950	eucJP	954, 5050, 33722	eucKR	970	eucTW	964	eucCN	1383	PCK	943	GBK	1386	koi8-r	878
Codeset name	Use CCSID																																						
ISO 8859-1	819 (default)																																						
ISO 8859-2	912																																						
ISO 8859-3	913																																						
ISO 8859-5	915																																						
ISO 8859-6	1089																																						
ISO 8859-7	813																																						
ISO 8859-8	916																																						
ISO 8859-9	920																																						
ISO 8859-13	921																																						
ISO 8859-15	923																																						
big5	950																																						
eucJP	954, 5050, 33722																																						
eucKR	970																																						
eucTW	964																																						
eucCN	1383																																						
PCK	943																																						
GBK	1386																																						
koi8-r	878																																						
Minimum Connections (mq_minConx)	<p>The minimum number of connections this adapter instance will maintain. When the adapter starts, it will pre-connect (“spin-up”) this many connections. Connections above this minimum, if not used for some time, will be closed and disconnected until the minimum is reached. Required. Valid value is 0-maximum MQ connections. Default is 0. Must be equal to or less than the maximum connections.</p>																																						
Maximum Connections (mq_maxConx)	<p>The maximum number of connections this adapter instance will attempt to use. Required. Valid value is 1-maximum MQ connections. Must be equal to or greater than the minimum connections. Default is 10.</p>																																						
This configuration will be used for: (mq_action)	<p>The type of configuration/action this instance will be used for. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ Sending messages to WebSphere MQ (MQ_SEND) ◆ Receiving messages from WebSphere MQ Sync (MQ_RCVSYNC) ◆ Receiving message from WebSphere MQ Async (MQ_RCVASYNC) <p>Note: There is no default value as it must be explicitly specified but the initial value in the UI is Sending messages to WebSphere MQ (MQ_SEND).</p>																																						

Field	Description
Open with MQOO_FAIL_IF QUIESCING option (mq_MQOO_failifquiescing)	Indicates whether or not to include the MQOO_FAIL_IF QUIESCING Open option. Required. Valid values are Yes (true) and No (false). Default is Yes (true).
Open with MQOO_SET_IDENTITY_CONTEXT option (mq_MQOO_setidentitycontext)	Indicates whether or not to include the MQOO_SET_IDENTITY_CONTEXT Open option. This option is only valid for send-type configurations and is used to allow setting the MQPMO_SET_IDENTITY_CONTEXT option. Required. Valid values are Yes (true) and No (false). Default is No (false).
Open with MQOO_SET_ALL_CONTEXT option (mq_MQOO_setallcontext)	Indicates whether or not to include the MQOO_SET_ALL_CONTEXT Open option. This option is only valid for send-type configurations and is used to allow setting the MQPMO_SET_ALL_CONTEXT option. Required. Valid values are Yes (true) and No (false). Default is No (false).
Turn on debug messages? (mq_debug)	Used to turn on debugging messages for this adapter instance. Required. Valid values are Yes (true) and No (false). Default is No (false).
Document Name (rcv_docName)	The document name to associate with the data received. If nothing is specified, the default "%^.dat" is used which generates a unique document name in the format yyyyymmddhhmmsslll.dat. Optional – uses default if not specified in the UI or GPM. Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.
MQGMO_SYNCPOINT (rcv_MQGMO_syncpoint)	Select Yes (true) to include this Get Message option. Optional – uses default if not specified in the GPM. Valid values are Yes (true) and No (false). Default is Yes (true). Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.
MQGMO_FAIL_IF QUIESCING (rcv_MQGMO_failifquiescing)	Select Yes (true) to include this Get Message option. Optional – uses default if not specified in the GPM. Valid values are Yes (true) and No (false). Default is Yes (true). Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.
MQGMO_LOGICAL_ORDER (rcv_MQGMO_logicalorder)	Select Yes (true) to include this Get Message option. Optional – uses default if not specified in the GPM. Valid values are Yes (true) and No (false). Default is Yes (true). Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.

Field	Description
MQGMO_ALL_SEGMENTS_AVAILABLE (rcv_MQGMO_allsegavail)	<p>Select Yes (true) to include this Get Message option. Optional – uses default if not specified in the GPM. Valid values are Yes (true) and No (false). Default is Yes (true).</p> <p>Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.</p>
MQGMO_ALL_MSGS_AVAILABLE (rcv_MQGMO_allmsgavail)	<p>Select Yes (true) to include this Get Message option. Optional – uses default if not specified in the GPM. Valid values are Yes (true) and No (false). Default is Yes (true).</p> <p>Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.</p>
MQGMO_COMPLETE_MSG (rcv_MQGMO_completemsg)	<p>Select Yes (true) to include this Get Message option. Optional – uses default if not specified in the GPM. Valid values are Yes (true) and No (false). Default is Yes (true).</p> <p>Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.</p>
MQGMO_CONVERT (rcv_MQGMO_convert)	<p>Select Yes (true) to include this Get Message option. Optional; uses default if not specified in the GPM. Valid values are Yes (true) and No (false). Default is No (false).</p> <p>Note: This field is only displayed if this is a Receive Async configuration. For Send and Receive Sync, set this in the GPM.</p>
Message Handling (rcv_msgHandling)	<p>Type of message handling to be used. Optional; uses default if not specified. Valid values are:</p> <ul style="list-style-type: none"> ◆ Receive one message as one document – receives one complete message, including any segments ◆ Receive all messages in a logical group as one document – receives all messages that are part of a logical group ◆ Receive all messages available as one document – receives any and all messages available ◆ Combine all messages with the same msgID and trigger using msgType – groups messages with the same msgID and with msgType of APPL_FIRST and continues to do so until msgType equals APPL_LAST <p>Default is Receive one message as one document.</p> <p>Note: This field is only displayed if this is a Receive Async configuration. For Receive Sync, set this in the GPM.</p>
Bootstrap Workflow (rcv_workFlowName_bootstrap)	<p>The business process name to start when data is received. Required if using Receive Async, otherwise optional.</p> <p>Note: This field is only displayed if this is a Receive Async configuration. For Receive Sync, set this in the GPM.</p>

Field	Description
Async Failure Workflow (rcv_workFlowName_failure)	When running in Async Receive mode, if something happens to the connection, a number of attempts will be made to re-establish the connection but if those fail, this option is a way to start a business process to perform notification of the problem. The adapter will need to be restarted once the problem is resolved.
Document Storage Type (docStorageType)	<p>Defines how the document will be stored in the system. Required. Valid values are:</p> <ul style="list-style-type: none"> ◆ System Default ◆ Database ◆ File System <p>Default is System Default.</p> <p>Note: For more information about document storage types, see <i>Managing Services and Adapters</i>.</p>
Buffer Size (buffersize)	<p>This is an optional parameter that only needs to be configured in certain circumstances. Valid values are 1 - 999999999, but smaller than 5120 is not recommended.</p> <p>When receiving, the default behavior is to use a buffer size that matches the size of the message being received. If you experience Out Of Memory issues when receiving very large messages from MQ (and not enough memory is available), set this buffer size appropriately in accordance with the size of messages you receive and how much memory you have available for use. Setting this value to 0 has the same effect as not setting it at all - the default behavior is used.</p> <p>When sending, this parameter can be used for performance tuning to increase throughput. The default buffer size is 5120 bytes (5k) if not overridden in the GPM or in BPML.</p> <p>Note: This field is only displayed if this is a Receive Async configuration. For Receive Sync or Send, set this in the GPM or in the BPML.</p>
Failure retry attempts (rcv_retryCount)	<p>The number of retry attempts when exceptions occur. Optional – uses default if not specified. Valid value is any valid integer value. The value -1 specifies infinite retry attempts. The value 0 specifies no retry attempts.</p> <p>Note: This field is only valid if this is a Receive Async configuration.</p>
Delay between retries (rcv_retrySleep)	<p>The number of milliseconds to wait before retrying. Optional – uses default if not specified. Valid value is any valid integer value. Default is 300000 milliseconds (5 minutes).</p> <p>Note: This field is only valid if this is a Receive Async configuration.</p>
Group Messages (rcv_groupBy)	<p>Which identifier to use when grouping messages. Only used if Message Handling is set to <i>Receive all messages in a logical group as one document</i>. Options are:</p> <ul style="list-style-type: none"> ◆ With the same group identifier ◆ With the same message identifier ◆ With the same correlation identifier.

Field	Description
Use the group status flag to determine end-of-group? (rcv_useGroupStatus)	Whether to use the group status to determine end of a group or to receive until no more messages are available. Only used if Message Handling is set to <i>Receive all messages in a logical group as one document</i> . Valid values: <ul style="list-style-type: none"> ◆ Yes ◆ No
Group messages even when the identifier is MQ*_NONE? (rcv_groupMsgWhenIdNone)	Whether to group messages when the identifier equals MQ*_NONE. Only used if Message Handling is set to <i>Receive all messages in a logical group as one document</i> . Valid values are: <ul style="list-style-type: none"> ◆ Yes ◆ No
User	User ID to associate with business processes that use this configuration. Enter the user ID or click the Filter icon, then select from the list.

GPM Configuration

The following table describes the fields used to configure the WebSphere MQ adapter in the GPM only. Fields that must be set in the Application configuration are not included in this table:

Note: Where the GPM values differ from the BPML values, the BPML values are shown in parentheses.

Field	Description
buffersize	<p>This is an optional parameter that only needs to be configured in certain circumstances. Valid values are 1 - 999999999, but smaller than 5120 is not recommended.</p> <p>When receiving, the default behavior is to use a buffer size that matches the size of the message being received. If you experience Out Of Memory issues when receiving very large messages from MQ (and not enough memory is available), set this buffer size appropriately in accordance with the size of messages you receive and how much memory you have available for use. Setting this value to 0 has the same effect as not setting it at all - the default behavior is used.</p> <p>When sending, this parameter can be used for performance tuning to increase throughput. The default buffer size is 5120 bytes (5k) if not overridden in the GPM or in BPML.</p> <p>Note: If this is a Receive Async configuration, this parameter is set in the Application configuration. For Receive Sync and Send configurations, you can also set this parameter in the BPML.</p>
rcv_docName	The document name to associate with the data received. Optional – uses default if not specified in the UI or GPM. If nothing is specified, the default “%^.dat” is used which generates a unique document name in the format <i>yyyymmddhhmmsslll.dat</i> .

Field	Description
rcv_groupBy	Which identifier to use when grouping messages. Only used if rcv_msgHandling is set to <i>Receive all messages in a logical group as one document</i> . Options are: <ul style="list-style-type: none"> ◆ With the same group identifier (GRPID) ◆ With the same message identifier (MSGID) ◆ With the same correlation identifier (CORID)
rcv_groupMsgWhenIdNone	Whether to group messages when the identifier equals MQ*_NONE. Valid values are: <ul style="list-style-type: none"> ◆ Yes (true) ◆ No (false)
rcv_HexToByte_corId	Select Yes if the value specified in <i>rcv_MQMO_corId</i> is a hex string that needs to be converted to a byte array for the MQ message header. Optional – uses default if not specified in the GPM. Valid values are Yes and No. Default is No.
rcv_HexToByte_grpId	Select Yes if the value specified in <i>rcv_MQMO_grpId</i> is a hex string that needs to be converted to a byte array for the MQ message header. Optional – uses default if not specified in the GPM. Valid values are Yes and No. Default is No.
rcv_HexToByte_msgId	Select Yes if the value specified in <i>rcv_MQMO_msgId</i> is a hex string that needs to be converted to a byte array for the MQ message header. Optional – uses default if not specified in the GPM. Valid values are Yes and No. Default is No.
rcv_MQGMO_wait	Select Yes (true) to include this Get Message option. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
rcv_MQGMO_waitInterval	The wait interval specified in 1/1000 sec. Must select Yes (true) on rcv_MQGMO_wait for this value to be used. Optional. Valid values are 1 - 999999999.
rcv_MQMO_corId	Sets the MQMO_MATCH_CORREL_ID option to this value. Optional.
rcv_MQMO_grpId	Sets the MQMO_MATCH_GROUP_ID option to this value. Optional.
rcv_MQMO_msgId	Sets the MQMO_MATCH_MSG_ID. Optional.
snd_HexToByte_actToken	Select Yes if the value specified in <i>snd_MQMD_accountingToken</i> is a hex string that needs to be converted to a byte array for the MQ message header. Optional – uses default if not specified in the GPM. Valid values are Yes and No. Default is No.
snd_HexToByte_corId	Select Yes if the value specified in <i>snd_MQMD_correlId</i> is a hex string that needs to be converted to a byte array for the MQ message header. Optional – uses default if not specified in the GPM. Valid values are Yes and No. Default is No.
snd_HexToByte_grpId	Select Yes if the value specified in <i>snd_MQMD_groupId</i> is a hex string that needs to be converted to a byte array for the MQ message header. Optional – uses default if not specified in the GPM. Valid values are Yes and No. Default is No.

Field	Description
snd_HexToByte_msgId	Select Yes if the value specified in <i>snd_MQMD_msgId</i> is a hex string that needs to be converted to a byte array for the MQ message header. Optional – uses default if not specified in the GPM. Valid values are Yes and No. Default is No.
snd_maxMsgSize	If the data being sent is larger than the size specified, the adapter will segment the message into messages of the specified size. If no value is given or if zero is specified, no segmenting is performed by the adapter. Optional. Valid values are 0 - 999999999.
snd_MQMD_accountingToken	Used to set the accounting token field of the message descriptor (MQMD). Optional.
snd_MQMD_applicationIdData	Used to set the application identity data field of the message descriptor (MQMD). Optional.
snd_MQMD_applicationOriginData	Used to set the application origin data field of the message descriptor (MQMD). Optional.
snd_MQMD_charset	Used to set the character set field of the message descriptor (MQMD). Optional – uses default if not specified. Valid value is a valid character set identifier as defined by the MQ documentation. Default is MQCCSI_Q_MGR, character set 819 (iso-8859-1/latin1/ibm819).
snd_MQMD_correlId	Used to set the correlation identifier field of the message descriptor (MQMD). Optional – uses default if not specified. Default is MQCI_NONE.
snd_MQMD_expiry	Used to set the expiry field of the message descriptor (MQMD). If zero is specified, MQEI_UNLIMITED is used. Optional – uses default if not specified. Valid values are 0 - 9999999999. Default is MQEI_UNLIMITED.
snd_MQMD_feedback	Used to set the feedback field of the message descriptor (MQMD). The feedback field is only set if <i>snd_msgType</i> is REPORT. Optional – uses default if not specified. Valid value is any valid feedback or reason code value. Default is MQFB_NONE. See the MQ documentation for more information.
snd_MQMD_format	Used to set the format field of the message descriptor (MQMD). Optional – uses default if not specified. Valid value is any valid format type. Default is MQFMT_NONE. See the MQ documentation for more information.
snd_MQMD_groupId	Used to set the group identifier field of the message descriptor (MQMD). Optional – uses default if not specified. Default is MQGI_NONE.
snd_MQMD_msgId	Used to set the message identifier field of the message descriptor (MQMD). Optional – uses default if not specified. Default is MQMI_NONE.
snd_MQMD_msgSeqNumber	Used to set the message sequence number field of the message descriptor (MQMD). Optional. Valid values are 1 - 999999999.

Field	Description
snd_MQMD_msgType	Used to set the message type field of the message descriptor (MQMD). Required when sending data. Valid values are: <ul style="list-style-type: none"> ◆ DATAGRAM ◆ REQUEST ◆ REPORT ◆ REPLY ◆ APPL_FIRST ◆ APPL_LAST
snd_MQMD_offset	Used to set the message offset field of the message descriptor (MQMD). Optional. Valid values are 0 - 999999999.
snd_MQMD_persistence	Used to set the message persistence field of the message descriptor (MQMD). Optional – uses default if not specified. Valid values are Yes, No, and QDEF. Default is MQPER_PERSISTENCE_AS_Q_DEF.
snd_MQMD_priority	Used to set the priority field of the message descriptor (MQMD). Optional – uses default if not specified. Valid values are 0 - 9. Default is MQPRI_PRIORITY_AS_Q_DEF.
snd_MQMD_putApplName	Used to set the put application name field of the message descriptor (MQMD). Optional.
snd_MQMD_putApplType	Used to set the put application type field of the message descriptor (MQMD). Optional. Valid value is a valid PutApplicationType value.
snd_MQMD_replyToQ	Used to set the reply to queue field of the message descriptor (MQMD). Optional.
snd_MQMD_replyToQM	Used to set the reply to queue manager field of the message descriptor (MQMD). Optional.
snd_MQMF_lastMsgInGroup	Used to add the MQMF_LAST_MSG_IN_GROUP Message Flag option of the message descriptor (MQMD). Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQMF_msgInGroup	Used to add the MQMF_MSG_IN_GROUP Message Flag option of the message descriptor (MQMD). Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQMF_segmentAllowed	Selecting yes adds the MQMF_SEGMENTATION_ALLOWED Message Flag option of the message descriptor (MQMD). Optional – uses default if not specified. Valid values are Yes and No. No sets MQMF_SEGMENTATION_INHIBITED.
snd_MQPMO_failIfQuietscing	Used to add the QMPMO_FAIL_IF QUIESCING Put Message option. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQPMO_logicalorder	Used to add the MQPMO_LOGICAL_ORDER Put Message option. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is Yes (true).

Field	Description
snd_MQPMO_newCorld	If yes, causes the queue manager to generate a new correlation identifier in the Correlld field of the message descriptor (MQMD). Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQPMO_newMsgld	If yes, causes the queue manager to generate a new message identifier in the Msgld field of the message descriptor (MQMD). Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQPMO_setallcontext	Used to add the MQPMO_SET_ALL_CONTEXT Put Message option which indicates that the user will supply all the identity and origin fields. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQPMO_setidentitycontext	Used to add the MQPMO_SET_IDENTITY_CONTEXT Put Message option which indicates that the user will supply all the identity fields. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQPMO_syncpoint	If yes, adds MQPMO_SYNCPOINT Put Message option. If no, adds MQPMO_NO_SYNCPOINT Put Message option. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is Yes (true).
snd_MQRO_coa	Sets the confirmation on arrival Report option. Optional – uses default if not specified. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA) Default is No Report (NOT_USED).
snd_MQRO_cod	Sets the confirmation on delivery Report option. Optional – uses default if not specified. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA) Default is No Report (NOT_USED).
snd_MQRO_discard	If yes, adds the MQRO_DISCARD_MSG Report option. Optional – uses default if not specified. Valid values are Yes and No. No assumes MQRO_DEAD_LETTER_Q.

Field	Description
snd_MQRO_exception	Sets the exception Report option. Optional – uses default if not specified. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA) Default is No Report (NOT_USED).
snd_MQRO_expiration	Sets the expiration Report option. Optional – uses default if not specified. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA) Default is No Report (NOT_USED).
snd_MQRO_nan	If yes, adds the MQRO_NAN Report option. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQRO_pan	If yes, adds the MQRO_PAN Report option. Optional – uses default if not specified. Valid values are Yes (true) and No (false). Default is No (false).
snd_MQRO_passCorId	If yes, adds the MQRO_PASS_CORREL_ID Report option. Optional – uses default if not specified. Valid values are Yes and No. No sets MQRO_COPY_MSG_ID_TO_CORREL_ID.
snd_MQRO_passMsgId	If yes, adds the MQRO_PASS_MSG_ID Report option. Optional – uses default if not specified. Valid values are Yes and No. No sets MQRO_NEW_MSG_ID.

Input from Business Process to Adapter

In addition to the parameters listed in the GPM configuration table, an optional document identifier can be passed to the WebSphere MQ adapter through the *MQ/documentId* parameter. The *MQ/documentId* parameter identifies a document to use instead of the primary document.

Another parameter that does not appear in the GPM but can be passed to the adapter, *snd_noSegSplit*, enables you to specify that a document should be broken into separate messages without segmentation. To turn this parameter on, specify the following in your BPML file: `<assign to="snd_noSegSplit">Yes</assign>`.

Output from Adapter to Business Process

The following tables identify parameters that are passed from the WebSphere MQ adapter to the business process.

After Sending

Field	Description
MQ/sentMsgId	The message identifier that was sent with the message. This value was either assigned by the user or assigned by the queue manager. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
MQ/sentCorrelationId	The correlation identifier that was sent with the message. This value was either assigned by the user or assigned by the queue manager. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
MQ/sentGroupId	The group identifier that was sent with the message. This value was either assigned by the user or assigned by the queue manager. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
MQ/nextMsgSeqNum	If sending multiple messages as part of a logical group but MQPMO_LOGICAL_ORDER is not used and MQMF_MSG_IN_GROUP and/or MQMF_LAST_MSG_IN_GROUP is specified, the adapter will automatically take care of incrementing the message sequence number from its initial value but only if segmenting is required. In this scenario, the value of this variable should be used in the snd_MQMD_msgSeqNumber parameter to set the next initial message sequence number.
MQ/nextMsgOffset	If sending multiple messages as part of a logical group but MQPMO_LOGICAL_ORDER is not used, the adapter will automatically take care of incrementing the message offset number from its initial value but only if segmenting is required. In this scenario, the value of this variable should be used in the snd_MQMD_offset parameter to set the next initial message offset value.

After Receiving

Field	Description
MQ/accountingToken	The accounting token of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
MQ/applicationIdData	The application identifier data of this message.
MQ/applicationOriginData	The application origin data of this message.
MQ/characterSet	The character set of this message.
MQ/correlationId	The correlation identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
MQ/document	If a message was received synchronously and did not start a new workflow (non-bootstrapped), the workflow document is created here instead of the primary document.
MQ/documentId	If the receive was performed synchronously and non-bootstrapped, this document identifier refers to the document received.
MQ/format	The format of this message.
MQ/groupId	The group identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.

Field	Description
MQ/msgExpiry	The message expiry of this message.
MQ/msgFeedback	The message feedback of this message.
MQ/msgFlags	The message flags of this message.
MQ/msgGroupStatus	The message group status of this message.
MQ/msgId	The message identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
MQ/msgLength	The message length of this message (if part of a segmented group – the length of the last segment).
MQ/msgSeqNum	The message sequence number of this message.
MQ/msgTotalLength	The total message length of all messages received if segmented or grouped.
MQ/msgType	The type of message received.
MQ/putApplicationName	The put application name of this message.
MQ/putApplType	The put application type of this message.
MQ/putDateTime	The put date and time of this message in the format “MM-dd-yyyy HH:mm:ss”
MQ/replyToQ	The reply to queue of this message.
MQ/replyToQM	The reply to queue manager of this message.
MQ/report	The contents of the report field of this message.
MQ/userId	The user identifier of this message.

WebSphereMQ Suite Async Receiver Adapter (Build 4300 - Build 4306)

The WebSphereMQ Suite Async Receiver adapter enables you to receive messages as soon as they are available on the queue instead of waiting for a scheduled job to poll the queue. The following table provides an overview of the WebSphereMQ Suite Async Receiver adapter:

System Name	WSMQAsyncRcv
Graphical Process Modeler (GPM) categories	None – cannot be used as part of a business process.
Description	Used to receive messages asynchronously and invoke business processes. Note: You need to configure this adapter if you are receiving CHIPS messages using the MQ transport mode in the CHIPS adapter.
Business usage	To receive messages as soon as they are available on the queue instead of waiting for a scheduled job to poll the queue.
Usage example	Once an instance of this adapter is configured, it immediately and continuously performs “gets” on the queue to receive messages and then starts the specified business process.
Preconfigured?	No.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms.
Related services	WSMQOpenSession, WSMQCloseSession, WSMQOpenQueue, WSMQCloseQueue, WSMQGetMessage, WSMQCommit, WSMQBackout.
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	Yes. Always invokes a business process when data is received.
Invocation	Cannot be invoked by a business process; runs as soon as it is configured.
Business process context considerations	This adapter can either automatically commit the messages received or it can pass the session identifier on to the invoked business process where further processing (puts, gets, etc) can occur. If “Get Type” is configured as “Get One”, the invoked business process will only contain the primary document. If “Get Type” is configured as “Get All”, the invoked business process will contain one or more documents based on the other message handling parameters. Note: The WSMQ Suite Commit service, which is the “other end” of this service, is ignored if it exists within a distributed transaction.
Returned status values	No status values returned since it cannot be used in a business process.
Restrictions	None.
Persistence level	Default.

Testing considerations	While testing this service, it is recommended that you turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.
------------------------	---

Implementing the WebSphereMQ Suite Async Receiver Adapter

To implement the WebSphereMQ Suite Async Receiver adapter, complete the following tasks:

1. Create a configuration of the WebSphereMQ Suite Async Receiver adapter.
2. Specify field settings for the adapter configuration in the Application Admin Console as necessary.

Configuring the WebSphereMQ Suite Async Receiver Adapter

You must specify field settings in Application, using the Admin Console.

Creating or Setting Up a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the WebSphereMQ Suite Async Receiver adapter, or to set up the configuration provided with Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – Do not include the configuration in a adapter group at this time.◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other adapters to the group as well.)◆ Select Group – If adapter groups already exist for this adapter type, they are displayed in the list. Select a group from the list. Note: See <i>Managing Services and Adapters</i> .
Host Name (wsmq_hostname)	The host name or IP address of the WebSphereMQ server. Valid value is alphanumeric. If not specified, bindings mode is used. Optional. Note: This parameter is required when configuring the adapter for use with CHIPS. This Host Name must be the same as the Websphere MQ Server Name that you configured in the CHIPS adapter.
Port Number (wsmq_port)	The listening port of the WebSphereMQ server. Valid value is a valid numeric port. Default is 1414. Optional. Note: This parameter is required when configuring this adapter for use with CHIPS. This Listening Port must be the same as the Websphere MQ Server Port No. that you configured in the CHIPS adapter.

Field	Description																																						
Queue Manager (wsmq_qmanager)	<p>The Queue Manager name to use. Valid value is alphanumeric. If not specified, uses the default queue manager. Optional.</p> <p>Note: This parameter is required when configuring this adapter for use with CHIPS. This Queue Manager name must be the same as the Reply-To Queue Manager name that you configured in the CHIPS adapter.</p>																																						
Channel (wsmq_channel)	<p>The channel to use. Valid value is alphanumeric. Required.</p> <p>Note: This parameter is required when configuring the adapter for use with CHIPS. This Channel name must be the same as the Channel name you configured in the CHIPS Adapter.</p>																																						
User Identifier (wsmq_userid)	<p>A user identifier if required to access the WebSphereMQ server. Valid value is alphanumeric. Optional.</p> <p>Note: This parameter is optional when configuring this adapter for use with CHIPS. This User Identifier must be the same as the Websphere MQ Server User ID that you configured in the CHIPS adapter, if present.</p>																																						
CCSID (mq_ccsid)	<p>If needed, enter the Coded Character Set Identifier (CCSID) that represents the codeset name you wish to use. Optional.</p> <table border="1"> <thead> <tr> <th>Codeset name</th> <th>Use CCSID</th> </tr> </thead> <tbody> <tr><td>ISO 8859-1</td><td>819 (default)</td></tr> <tr><td>ISO 8859-2</td><td>912</td></tr> <tr><td>ISO 8859-3</td><td>913</td></tr> <tr><td>ISO 8859-5</td><td>915</td></tr> <tr><td>ISO 8859-6</td><td>1089</td></tr> <tr><td>ISO 8859-7</td><td>813</td></tr> <tr><td>ISO 8859-8</td><td>916</td></tr> <tr><td>ISO 8859-9</td><td>920</td></tr> <tr><td>ISO 8859-13</td><td>921</td></tr> <tr><td>ISO 8859-15</td><td>923</td></tr> <tr><td>big5</td><td>950</td></tr> <tr><td>eucJP</td><td>954, 5050, 33722</td></tr> <tr><td>eucKR</td><td>970</td></tr> <tr><td>eucTW</td><td>964</td></tr> <tr><td>eucCN</td><td>1383</td></tr> <tr><td>PCK</td><td>943</td></tr> <tr><td>GBK</td><td>1386</td></tr> <tr><td>koi8-r</td><td>878</td></tr> </tbody> </table> <p>Note: CCSID is not used when connecting directly using "binding mode."</p>	Codeset name	Use CCSID	ISO 8859-1	819 (default)	ISO 8859-2	912	ISO 8859-3	913	ISO 8859-5	915	ISO 8859-6	1089	ISO 8859-7	813	ISO 8859-8	916	ISO 8859-9	920	ISO 8859-13	921	ISO 8859-15	923	big5	950	eucJP	954, 5050, 33722	eucKR	970	eucTW	964	eucCN	1383	PCK	943	GBK	1386	koi8-r	878
Codeset name	Use CCSID																																						
ISO 8859-1	819 (default)																																						
ISO 8859-2	912																																						
ISO 8859-3	913																																						
ISO 8859-5	915																																						
ISO 8859-6	1089																																						
ISO 8859-7	813																																						
ISO 8859-8	916																																						
ISO 8859-9	920																																						
ISO 8859-13	921																																						
ISO 8859-15	923																																						
big5	950																																						
eucJP	954, 5050, 33722																																						
eucKR	970																																						
eucTW	964																																						
eucCN	1383																																						
PCK	943																																						
GBK	1386																																						
koi8-r	878																																						

Field	Description
Queue Name (wmq_qname)	<p>The name of a previously opened queue used to PUT messages. Any value is valid. Required.</p> <p>Note: This parameter is required when configuring for use with CHIPS. This Queue Name must be the same as the Reply-To Queue name that you configured in the CHIPS adapter.</p>
Remote Queue Manager	<p>The name of the remote queue manager, which is any queue manager other than the local queue manager. A remote queue manager exists on a remote machine across the network, or on the same machine as the local queue manager. Optional.</p>
Dynamic Queue Name	<p>The name of the Dynamic Queue. Dynamic queues are created by the queue manager when an application issues an MQOPEN request specifying a queue name that is the name of a model queue. The dynamic queue that is created in this way is a local queue whose attributes are taken from the model queue definition. Optional.</p> <p>Note: The Dynamic Queue Name can be specified by the application or the queue manager can generate the name and return it to the application. Dynamic queues defined in this way are either temporary queues, which do not survive product restarts, or permanent queues that do survive.</p>
Alternate User Identifier	<p>The identifier for the alternate user. Alternate-user authority controls whether one user profile can use the authority of another user profile when accessing a Web Sphere MQ object. Optional.</p> <p>Note: This authority is essential when a server receives requests from a program and the server wants to ensure that the program has the required authority for the request. The server might have the required authority, but it needs to know whether the program has the authority for the actions it has requested.</p>
Binding Options (wsmq_MQOO_binding)	<p>These options apply when the queue being opened is a cluster queue. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ MQOO_BIND_AS_Q_DEF (ASQDEF) (default) ◆ MQOO_BIND_ON_OPEN (ONOPEN) ◆ MQOO_BIND_NOT_FIXED (NOTFIXED)
Context Options (wsmq_MQOO_context)	<p>These options control the processing of message context. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ No context setting (NONE) (default) ◆ MQOO_SET_IDENTITY_CONTEXT (IDCTX) ◆ MQOO_SET_ALL_CONTEXT (ALLCTX)
MQOO_FAIL_IF QUIESCING? (wsmq_MQOO_failifquiescing)	<p>Indicates whether or not to include the MQOO_FAIL_IF QUIESCING Open Option. Valid values are Yes and No. Default is Yes. Optional.</p>

Field	Description
Automatically Commit? (wsmq_autocommit)	Determines if the adapter should automatically commit messages received or pass on the session identifier to the invoked business process for additional processing. Valid values are Yes and No. Default is Yes. Optional.
Enable debug messages? (wsmq_debug)	Used to turn on debugging messages for this adapter instance. Valid values are Yes and No. Default is No. Optional.
Get Type (wsmq_type)	Specifies the type of Get to perform. Optional. Valid values are: <ul style="list-style-type: none"> ◆ GETONE (default) ◆ GETALL ◆ GETUNDERCURSOR ◆ BROWSEALL ◆ BROWSENEXT ◆ BROWSEFIRST ◆ BROWSEUNDERCURSOR <p>Note: This parameter is required when configuring this adapter for use with CHIPS, and the required Get Type is GETONE.</p>
Receive Message Limit (wsmq_rcvMsgLimit)	Used with GETALL or BROWSEALL to limit the number of messages received. Valid values are 0 (unlimited) to 999999999. Optional.
MQGMO_ALL_MSGS_AVAILABLE (wsmq_MQGMO_allmsgavail)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.
MQGMO_ALL_SEGMENTS_AVAILABLE (wsmq_MQGMO_allsegavail)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.
MQGMO_COMPLETE_MSG (wsmq_MQGMO_completemsg)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.
MQGMO_CONVERT (wsmq_MQGMO_convert)	Valid values are Yes and No. Default is No. Select Yes to include this Get Message Option. Optional.
MQGMO_FAIL_IF QUIESCING (wsmq_MQGMO_failifquiescing)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.
MQGMO_LOGICAL_ORDER (wsmq_MQGMO_logicalorder)	Select Yes (default) to include this Get Message Option. Optional.
MQGMO_SYNCPOINT (wsmq_MQGMO_syncpoint)	Select Yes (default) to include this Get Message Option. Optional.
Wait Interval (milliseconds) (wsmq_MQGMO_waitInterval_async)	Specifies the polling interval. Should be less than 15 minutes, that is, session timeout. Valid value is any valid long integer value less than session timeout of 15 minutes (900000 milliseconds). Optional.

Field	Description
Message Handling (wsmq_msgHandling)	Select the type of message handling to be used. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Receive one message as one document (ONE) (default) ◆ Receive all messages in a logical group as one document (GROUP) ◆ Receive all messages available as one document (ALL) ◆ Combine all messages with the same msgID and trigger using msgType (SPEC1)
Group messages (wsmq_groupBy)	Selects which identifier to use when grouping messages. Optional. Valid values are: <ul style="list-style-type: none"> ◆ With the same group identifier (GRPID) ◆ With the same message identifier (MSGID) ◆ With the same correlation identifier (CORID) <p>Note: Only displayed if the wsmq_msgHandling parm is set to GROUP.</p>
Use the group status flag to determine end-of-group? (wsmq_useGroupStatus)	Specifies whether to use the group status to determine end of a group or to receive until no more messages are available. Valid values are Yes and No. Default is No. Optional. <p>Note: Only displayed if the wsmq_msgHandling parm is set to GROUP.</p>
Group messages even when the identifier is MQ*_NONE? (wsmq_groupMsgWhenIdNone)	Determines if messages will be grouped even if the identifier equals MQ*_NONE. Valid values are Yes and No Default is Yes. Optional. <p>Note: Only displayed if the wsmq_msgHandling parm is set to GROUP.</p>
Document Storage Type (docStorageType)	Determines the media used for temporary storage of data while processing. Optional. Valid values are: <ul style="list-style-type: none"> ◆ System Default (sd) (default) ◆ Database (db) ◆ File System (fs)
Document tracking (wsmq_docTracking)	Specifies whether to perform document tracking. Valid values are Yes and No. Default is No. Optional.
Document Name (wsmq_docName)	The document name to associate with the data received. Any value is valid. Default is %^.dat. Optional.
Buffersize override (wsmq_buffersize)	If specified, overrides the default buffersize used when streaming data. Valid values are 0-999999999. Optional.

Field	Description
Metadata1 To Include (wmq_metadata1)	<p>Specifies which metadata fields (added together if multiple) from the message to include with the Document created in ProcessData. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 1=msgId ◆ 2=corId ◆ 4=grpId ◆ 8=msgType ◆ 16=replyToQM ◆ 32=replyToQ ◆ 64=acctToken ◆ 128=ApplIdData ◆ 256=ApplOrigin ◆ 512=Format ◆ 1024=Report ◆ 2048=Feedback
Metadata2 To Include (wmq_metadata2)	<p>Specifies which metadata fields (added together if multiple) from the message to include with the Document created in ProcessData. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 1=groupStatus ◆ 2=encoding ◆ 4=charset ◆ 8=expiry ◆ 16=putDateTime ◆ 32=putApplName ◆ 64=putApplType ◆ 128=msgFlags ◆ 256=msgSeqNum ◆ 512=offset ◆ 1024=persistence ◆ 2048=priority
Maximum Bootstrap Threads (wmq_maxThreads)	<p>The maximum number of threads to use when invoking business processes. Valid values are 0 (unlimited) - 99999. Optional.</p>

Field	Description
Bootstrap Workflow (wsmq_bootFlow)	<p>The name of the business process to invoke once messages are received. Valid value is an existing business process. Required.</p> <p>Note: This parameter is required when configuring the adapter for use with CHIPS. If you are using MQ as a transport method for CHIPS, you must select the CHIPSUtility_ReceiveHandler.bpml business process. This preloaded system business process checks the Queue Manager Name, Queue name and Channel Name and uses the correct CHIPS Adapter to send back the required acknowledgement. The Queue Manager name, Queue Name, and Channel Name are taken from the Process Data when this business process is invoked. Based on the values from the Process Data, it will check the values in the CHIPS adapter Reply-To Queue Manager, Reply-To Queue, and Channel Name to pick up the correct CHIPS adapter instance.</p>
Failure Workflow (wsmq_failFlow)	<p>Specifies the business process to start if all failure retries have been exhausted. Valid value is a valid and existing business process. Optional.</p>
Failure retry attempts wsmq_retryCount()	<p>The number of retry attempts when exceptions occur. Valid value is any valid integer. Using the value “-1” specifies infinite retry attempts. 0 specifies no retry attempts. Optional.</p>
Delay between retries (wsmq_retrySleep)	<p>The number of milliseconds to wait before retrying. Valid value is any valid long integer. Optional.</p>
User (wsmq_userID)	<p>This user identifier is used as part of the security context for the invoked business process. Optional.</p> <p>Note: This parameter is optional when configuring for CHIPS. This User Identifier must be the same as the Websphere MQ Server User ID configured in the CHIPS adapter, if present.</p>
Password (wsmq_password)	<p>A user password if required to access the WebSphereMQ server. Valid value is alphanumeric. Optional.</p> <p>Note: This parameter is optional when configuring for use with CHIPS. This Password must be the same as the Websphere MQ Server Password that you configured in the CHIPS adapter, if present.</p>

Parameters Passed From Adapter to Invoked Business Process

The following table contains the parameters passed from the WebSphereMQ Suite Async Receiver adapter to the business process it invokes:

Parameter	Description
wsmq_sessionid	This value is only passed to the invoked business process if wsmq_autocommit=No. The invoked business process is then responsible for making sure that the data is committed or backed out as necessary and that the session is properly closed. Optional. See the example that follows this table.
accountingToken	The accounting token of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
applicationIdData	The application identifier data of this message
applicationOriginData	The application origin data of this message
characterSet	The character set of this message
correlationId	The correlation identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
DocumentX	If wsmq_type=GETONE, then PrimaryDocument is created. If wsmq_type=GETALL, one or more documents are created in sequential order starting with 1 (for example, Document1, Document2, Document3...).
DocumentCount	The number of documents created in ProcessData.
encoding	The message encoding.
expiry	The message expiry of this message.
feedback	The message feedback of this message.
format	The format of this message.
groupId	The group identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
groupStatus	The message group status of this message.
messageFlags	The message flags of this message
messageId	The message identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
messageSequenceNumber	The message sequence number of this message.
messageType	The type of message received.
offset	The current message offset.
persistence	The persistence setting for this message.
priority	The priority setting for this message.
putApplicationName	The put application name of this message.

Parameter	Description
putApplicationType	The put application type of this message.
putDateTime	The put date and time of this message in the format "MM-dd-yyyy HH:mm:ss"
replyToQueueManagerName	The reply to queue manager of this message.
replyToQueueName	The reply to queue of this message.
report	The contents of the report field of this message.

Examples

Example 1 – Process Data

The values specified for `wsmq_metadata1` and `wsmq_metadata2` determine which metadata fields are included with each document. The following example is what process data would look like as a result of specifying 4095 for both `wsmq_metadata1` and `wsmq_metadata2`:

```
<ProcessData>
  <WSMQ>
    <DocumentCount>1</documentCount>
    <Document1 SCIObjectID="df8f5e:102fa5a6c8f:-7414">
      <messageId>414D5120514D5F6761727931303030205D624C4220000502</messageId>
      <correlationId>0000000000000000000000000000000000000000000000000000000000000000</correlationId>
      <groupId>0000000000000000000000000000000000000000000000000000000000000000</groupId>

    <accountingToken>16010515000000B5E512BBA14EC030000000000000000B</accountingToken>
      <replyToQueueManagerName>QM_gary1000
    </replyToQueueManagerName>
      <replyToQueueName/>
      <applicationIdData/>
      <applicationOriginData/>
      <messageType verbose="Datagram">8</messageType>
      <format>MQSTR    </format>
      <report>0</report>
      <feedback>0</feedback>
      <groupStatus/>
      <encoding>273</encoding>
      <characterSet>819</characterSet>
      <expiry>-1</expiry>
      <putDateTime>03-31-2005 15:58:18</putDateTime>
      <putApplicationName>MQSeries Client for Java    </putApplicationName>
      <putApplicationType>28</putApplicationType>
      <messageFlags>0</messageFlags>
      <messageSequenceNumber>1</messageSequenceNumber>
      <offset>0</offset>
      <persistence>0</persistence>
      <priority>0</priority>
    </Document1>
  </WSMQ>
</ProcessData>
```

Example 2 – Processing Batch Documents Using GETALL

The following is an example of how to process batch documents received from a WebSphereMQ server. The following example could be used as the bootstrap workflow defined in the WebSphereMQ Suite Async Rcv adapter. This example iterates through all batch documents received and extracts each of them using the File System adapter:

```
<process name="(MyBusinessProcess)">
  <rule name="has more docs">
    <condition>number(/ProcessData/index/text()) <
number(/ProcessData/WSMQ/DocumentCount/text())</condition>
  </rule>

  <sequence>
    <assign to="index">0</assign>
    <choice name="for each doc">
      <select>
        <case ref="has more docs" activity="process doc"/>
      </select>

      <sequence name="process doc">
        <assign to="index" from="number(number(index) + 1)"></assign>
        <operation name="File System Adapter">
          <participant name="(My_File_System_Adapter)"/>
          <output message="FileSystemInputMessage">
            <assign to="." from="*"></assign>
            <assign to="Action">FS_EXTRACT</assign>
            <assign to="PrimaryDocument"
from="/ProcessData/WSMQ/*[number(/ProcessData/index/text())]/@SCIOBJECTID"></assign>

          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>

        <repeat name="next doc" ref="for each doc"/>
      </sequence>
    </choice>
  </sequence>
</process>
```

WebSphereMQ Suite Async Receiver Adapter (Build 4307 or higher)

The WebSphereMQ Suite Async Receiver adapter enables you to receive messages as soon as they are available on the queue instead of waiting for a scheduled job to poll the queue. The following table provides an overview of the WebSphereMQ Suite Async Receiver adapter:

System Name	WSMQAsyncRcv
Graphical Process Modeler (GPM) categories	None – cannot be used as part of a business process.
Description	Used to receive messages asynchronously and invoke business processes. Note: You need to configure this adapter if you are receiving CHIPS messages using the MQ transport mode in the CHIPS adapter.
Business usage	To receive messages as soon as they are available on the queue instead of waiting for a scheduled job to poll the queue.
Usage example	Once an instance of this adapter is configured, it immediately and continuously performs “gets” on the queue to receive messages and then starts the specified business process.
Preconfigured?	No.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported application platforms.
Related services	WSMQOpenSession, WSMQCloseSession, WSMQOpenQueue, WSMQCloseQueue, WSMQGetMessage, WSMQCommit, WSMQBackout.
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	Yes. Always invokes a business process when data is received.
Invocation	Cannot be invoked by a business process; runs as soon as it is configured.
Business process context considerations	This adapter can either automatically commit the messages received or it can pass the session identifier on to the invoked business process where further processing (puts, gets, etc) can occur. If “Get Type” is configured as “Get One”, the invoked business process will only contain the primary document. If “Get Type” is configured as “Get All”, the invoked business process will contain one or more documents based on the other message handling parameters. Note: The WSMQ Suite Commit service, which is the “other end” of this service, is ignored if it exists within a distributed transaction.
Returned status values	No status values returned since it cannot be used in a business process.
Restrictions	None.
Persistence level	Default.

Testing considerations	While testing this service, it is recommended that you turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.
------------------------	---

Implementing the WebSphereMQ Suite Async Receiver Adapter

To implement the WebSphereMQ Suite Async Receiver adapter, complete the following tasks:

1. Create a configuration of the WebSphereMQ Suite Async Receiver adapter.
2. Specify field settings for the adapter configuration in the Application Admin Console as necessary.

Configuring the WebSphereMQ Suite Async Receiver Adapter

You must specify field settings in Application, using the Admin Console.

Creating or Setting Up a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the WebSphereMQ Suite Async Receiver adapter, or to set up the configuration provided with Application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – Do not include the configuration in a adapter group at this time. ◆ Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other adapters to the group as well.) ◆ Select Group – If adapter groups already exist for this adapter type, they are displayed in the list. Select a group from the list. <p>Note: See <i>Managing Services and Adapters</i>.</p>
Host Name (wsmq_hostname)	<p>The host name or IP address of the WebSphereMQ server. Valid value is alphanumeric. If not specified, bindings mode is used. Optional.</p> <p>Note: This parameter is required when configuring the adapter for use with CHIPS. This Host Name must be the same as the Websphere MQ Server Name that you configured in the CHIPS adapter.</p>
Port Number (wsmq_port)	<p>The listening port of the WebSphereMQ server. Valid value is a valid numeric port. Default is 1414. Optional.</p> <p>Note: This parameter is required when configuring this adapter for use with CHIPS. This Listening Port must be the same as the Websphere MQ Server Port No. that you configured in the CHIPS adapter.</p>

Field	Description																																						
Queue Manager (wsmq_qmanager)	<p>The Queue Manager name to use. Valid value is alphanumeric. If not specified, uses the default queue manager. Optional.</p> <p>Note: This parameter is required when configuring this adapter for use with CHIPS. This Queue Manager name must be the same as the Reply-To Queue Manager name that you configured in the CHIPS adapter.</p>																																						
Channel (wsmq_channel)	<p>The channel to use. Valid value is alphanumeric. Required.</p> <p>Note: This parameter is required when configuring the adapter for use with CHIPS. This Channel name must be the same as the Channel name you configured in the CHIPS Adapter.</p>																																						
User Identifier (wsmq_userid)	<p>A user identifier if required to access the WebSphereMQ server. Valid value is alphanumeric. Optional.</p> <p>Note: This parameter is optional when configuring this adapter for use with CHIPS. This User Identifier must be the same as the Websphere MQ Server User ID that you configured in the CHIPS adapter, if present.</p>																																						
CCSID (mq_ccsid)	<p>If needed, enter the Coded Character Set Identifier (CCSID) that represents the codeset name you wish to use. Optional.</p> <table border="1"> <thead> <tr> <th>Codeset name</th> <th>Use CCSID</th> </tr> </thead> <tbody> <tr><td>ISO 8859-1</td><td>819 (default)</td></tr> <tr><td>ISO 8859-2</td><td>912</td></tr> <tr><td>ISO 8859-3</td><td>913</td></tr> <tr><td>ISO 8859-5</td><td>915</td></tr> <tr><td>ISO 8859-6</td><td>1089</td></tr> <tr><td>ISO 8859-7</td><td>813</td></tr> <tr><td>ISO 8859-8</td><td>916</td></tr> <tr><td>ISO 8859-9</td><td>920</td></tr> <tr><td>ISO 8859-13</td><td>921</td></tr> <tr><td>ISO 8859-15</td><td>923</td></tr> <tr><td>big5</td><td>950</td></tr> <tr><td>eucJP</td><td>954, 5050, 33722</td></tr> <tr><td>eucKR</td><td>970</td></tr> <tr><td>eucTW</td><td>964</td></tr> <tr><td>eucCN</td><td>1383</td></tr> <tr><td>PCK</td><td>943</td></tr> <tr><td>GBK</td><td>1386</td></tr> <tr><td>koi8-r</td><td>878</td></tr> </tbody> </table> <p>Note: CCSID is not used when connecting directly using "binding mode."</p>	Codeset name	Use CCSID	ISO 8859-1	819 (default)	ISO 8859-2	912	ISO 8859-3	913	ISO 8859-5	915	ISO 8859-6	1089	ISO 8859-7	813	ISO 8859-8	916	ISO 8859-9	920	ISO 8859-13	921	ISO 8859-15	923	big5	950	eucJP	954, 5050, 33722	eucKR	970	eucTW	964	eucCN	1383	PCK	943	GBK	1386	koi8-r	878
Codeset name	Use CCSID																																						
ISO 8859-1	819 (default)																																						
ISO 8859-2	912																																						
ISO 8859-3	913																																						
ISO 8859-5	915																																						
ISO 8859-6	1089																																						
ISO 8859-7	813																																						
ISO 8859-8	916																																						
ISO 8859-9	920																																						
ISO 8859-13	921																																						
ISO 8859-15	923																																						
big5	950																																						
eucJP	954, 5050, 33722																																						
eucKR	970																																						
eucTW	964																																						
eucCN	1383																																						
PCK	943																																						
GBK	1386																																						
koi8-r	878																																						

Field	Description
Queue Name (wmq_qname)	The name of a previously opened queue used to PUT messages. Any value is valid. Required. Note: This parameter is required when configuring for use with CHIPS. This Queue Name must be the same as the Reply-To Queue name that you configured in the CHIPS adapter.
Remote Queue Manager	The name of the remote queue manager, which is any queue manager other than the local queue manager. A remote queue manager exists on a remote machine across the network, or on the same machine as the local queue manager. Optional.
Dynamic Queue Name	The name of the Dynamic Queue. Dynamic queues are created by the queue manager when an application issues an MQOPEN request specifying a queue name that is the name of a model queue. The dynamic queue that is created in this way is a local queue whose attributes are taken from the model queue definition. Optional. Note: The Dynamic Queue Name can be specified by the application or the queue manager can generate the name and return it to the application. Dynamic queues defined in this way are either temporary queues, which do not survive product restarts, or permanent queues that do survive.
Alternate User Identifier	The identifier for the alternate user. Alternate-user authority controls whether one user profile can use the authority of another user profile when accessing a Web Sphere MQ object. Optional. Note: This authority is essential when a server receives requests from a program and the server wants to ensure that the program has the required authority for the request. The server might have the required authority, but it needs to know whether the program has the authority for the actions it has requested.
Binding Options (wsmq_MQOO_binding)	These options apply when the queue being opened is a cluster queue. Optional. Valid values are: <ul style="list-style-type: none"> ◆ MQOO_BIND_AS_Q_DEF (ASQDEF) (default) ◆ MQOO_BIND_ON_OPEN (ONOPEN) ◆ MQOO_BIND_NOT_FIXED (NOTFIXED)
Context Options (wsmq_MQOO_context)	These options control the processing of message context. Optional. Valid values are: <ul style="list-style-type: none"> ◆ No context setting (NONE) (default) ◆ MQOO_SET_IDENTITY_CONTEXT (IDCTX) ◆ MQOO_SET_ALL_CONTEXT (ALLCTX)
MQOO_FAIL_IF QUIESCING? (wsmq_MQOO_failifquiescing)	Indicates whether or not to include the MQOO_FAIL_IF QUIESCING Open Option. Valid values are Yes and No. Default is Yes. Optional.

Field	Description
Automatically Commit? (wsmq_autocommit)	Determines if the adapter should automatically commit messages received or pass on the session identifier to the invoked business process for additional processing. Valid values are Yes and No. Default is Yes. Optional.
Enable debug messages? (wsmq_debug)	Used to turn on debugging messages for this adapter instance. Valid values are Yes and No. Default is No. Optional.
SSL (SSL_SETTING_ssl_option)	Whether SSL is active. Valid values: <ul style="list-style-type: none"> ◆ Must (SSL_MUST) ◆ None (SSL_NONE)
Cipher Suite (SSL_SETTING_cipherSuite)	Specifies a valid SSL Version 2 or Version 3 cipher. For example, SSL_RSA_WITH_3DES_EDE_CBC_SHA.
Key Certificate (System Store) (SSL_SETTING_keyCertID)	Specifies a valid key certificate for client authentication. For example: frcppe03z3:1df073d:1153772f2cb:-66de mg2sdsb:1a679b7:116218d328a:-5e84
CA Certificate (SSL_SETTING_ca_cert_ids)	Specifies a CA certificate for server authentication. For MQ, only one certificate may be selected. For example: Measle1:1decdec:11159ba495b:-583c,frcppe03z3:1037c71:11584abf184:-7c4d,Measle1:1decdec:11159ba495b:-5837 Measle1:1decdec:11159ba495b:-5837
Get Type (wsmq_type)	Specifies the type of Get to perform. Optional. Valid values are: <ul style="list-style-type: none"> ◆ GETONE (default) ◆ GETALL ◆ GETUNDERCURSOR ◆ BROWSEALL ◆ BROWSENEXT ◆ BROWSEFIRST ◆ BROWSEUNDERCURSOR <p>Note: This parameter is required when configuring this adapter for use with CHIPS, and the required Get Type is GETONE.</p>
Receive Message Limit (wsmq_rcvMsgLimit)	Used with GETALL or BROWSEALL to limit the number of messages received. Valid values are 0 (unlimited) to 999999999. Optional.
MQGMO_ALL_MSGS_AVAILABLE (wsmq_MQGMO_allmsgavail)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.
MQGMO_ALL_SEGMENTS_AVAILABLE (wsmq_MQGMO_allsegavail)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.
MQGMO_COMPLETE_MSG (wsmq_MQGMO_completemsg)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.

Field	Description
MQGMO_CONVERT (wsmq_MQGMO_convert)	Valid values are Yes and No. Default is No. Select Yes to include this Get Message Option. Optional.
MQGMO_FAIL_IF QUIESCING (wsmq_MQGMO_failifquiescing)	Valid values are Yes and No. Select Yes (default) to include this Get Message Option. Optional.
MQGMO_LOGICAL_ORDER (wsmq_MQGMO_logicalorder)	Select Yes (default) to include this Get Message Option. Optional.
MQGMO_SYNCPOINT (wsmq_MQGMO_syncpoint)	Select Yes (default) to include this Get Message Option. Optional.
Wait Interval (milliseconds) (wsmq_MQGMO_waitInterval_async)	Specifies the polling interval. Should be less than 15 minutes, that is, session timeout. Valid value is any valid long integer value less than session timeout of 15 minutes (900000 milliseconds). Optional.
Message Handling (wsmq_msgHandling)	Select the type of message handling to be used. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Receive one message as one document (ONE) (default) ◆ Receive all messages in a logical group as one document (GROUP) ◆ Receive all messages available as one document (ALL) ◆ Combine all messages with the same msgID and trigger using msgType (SPEC1)
Group messages (wsmq_groupBy)	Selects which identifier to use when grouping messages. Optional. Valid values are: <ul style="list-style-type: none"> ◆ With the same group identifier (GRPID) ◆ With the same message identifier (MSGID) ◆ With the same correlation identifier (CORID) Note: Only displayed if the wsmq_msgHandling parm is set to GROUP.
Use the group status flag to determine end-of-group? (wsmq_useGroupStatus)	Specifies whether to use the group status to determine end of a group or to receive until no more messages are available. Valid values are Yes and No. Default is No. Optional. Note: Only displayed if the wsmq_msgHandling parm is set to GROUP.
Group messages even when the identifier is MQ*_NONE? (wsmq_groupMsgWhenIdNone)	Determines if messages will be grouped even if the identifier equals MQ*_NONE. Valid values are Yes and No Default is Yes. Optional. Note: Only displayed if the wsmq_msgHandling parm is set to GROUP.

Field	Description
Document Storage Type (docStorageType)	Determines the media used for temporary storage of data while processing. Optional. Valid values are: <ul style="list-style-type: none"> ◆ System Default (sd) (default) ◆ Database (db) ◆ File System (fs)
Document tracking (wsmq_docTracking)	Specifies whether to perform document tracking. Valid values are Yes and No. Default is No. Optional.
Document Name (wsmq_docName)	The document name to associate with the data received. Any value is valid. Default is %^.dat. Optional.
Buffersize override (wsmq_buffersize)	If specified, overrides the default buffersize used when streaming data. Valid values are 0-999999999. Optional.
Metadata1 To Include (wsmq_metadata1)	Specifies which metadata fields (added together if multiple) from the message to include with the Document created in ProcessData. Optional. Valid values are: <ul style="list-style-type: none"> ◆ 1=msgId ◆ 2=corId ◆ 4=grpId ◆ 8=msgType ◆ 16=replyToQM ◆ 32=replyToQ ◆ 64=acctToken ◆ 128=ApplIdData ◆ 256=ApplOrigin ◆ 512=Format ◆ 1024=Report ◆ 2048=Feedback

Field	Description
Metadata2 To Include (wmq_metadata2)	<p>Specifies which metadata fields (added together if multiple) from the message to include with the Document created in ProcessData. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 1=groupStatus ◆ 2=encoding ◆ 4=charset ◆ 8=expiry ◆ 16=putDateTime ◆ 32=putApplName ◆ 64=putApplType ◆ 128=msgFlags ◆ 256=msgSeqNum ◆ 512=offset ◆ 1024=persistence ◆ 2048=priority
Maximum Bootstrap Threads (wmq_maxThreads)	The maximum number of threads to use when invoking business processes. Valid values are 0 (unlimited) - 99999. Optional.
Bootstrap Workflow (wsmq_bootFlow)	<p>The name of the business process to invoke once messages are received. Valid value is an existing business process. Required.</p> <p>Note: This parameter is required when configuring the adapter for use with CHIPS. If you are using MQ as a transport method for CHIPS, you must select the CHIPSUtility_ReceiveHandler.bpml business process. This preloaded system business process checks the Queue Manager Name, Queue name and Channel Name and uses the correct CHIPS Adapter to send back the required acknowledgement. The Queue Manager name, Queue Name, and Channel Name are taken from the Process Data when this business process is invoked. Based on the values from the Process Data, it will check the values in the CHIPS adapter Reply-To Queue Manager, Reply-To Queue, and Channel Name to pick up the correct CHIPS adapter instance.</p>
Failure Workflow (wsmq_failFlow)	Specifies the business process to start if all failure retries have been exhausted. Valid value is a valid and existing business process. Optional.
Failure retry attempts wsmq_retryCount()	The number of retry attempts when exceptions occur. Valid value is any valid integer. Using the value “-1” specifies infinite retry attempts. 0 specifies no retry attempts. Optional.

Field	Description
Delay between retries (wsmq_retrySleep)	The number of milliseconds to wait before retrying. Valid value is any valid long integer. Optional.
User (wsmq_userID)	This user identifier is used as part of the security context for the invoked business process. Optional. Note: This parameter is optional when configuring for CHIPS. This User Identifier must be the same as the Websphere MQ Server User ID configured in the CHIPS adapter, if present.
Password (wsmq_password)	A user password if required to access the WebSphereMQ server. Valid value is alphanumeric. Optional. Note: This parameter is optional when configuring for use with CHIPS. This Password must be the same as the Websphere MQ Server Password that you configured in the CHIPS adapter, if present.

Parameters Passed From Adapter to Invoked Business Process

The following table contains the parameters passed from the WebSphereMQ Suite Async Receiver adapter to the business process it invokes:

Parameter	Description
wsmq_sessionid	This value is only passed to the invoked business process if wsmq_autocommit=No. The invoked business process is then responsible for making sure that the data is committed or backed out as necessary and that the session is properly closed. Optional. See the example that follows this table.
accountingToken	The accounting token of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
applicationIdData	The application identifier data of this message
applicationOriginData	The application origin data of this message
characterSet	The character set of this message
correlationId	The correlation identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
DocumentX	If wsmq_type=GETONE, then PrimaryDocument is created. If wsmq_type=GETALL, one or more documents are created in sequential order starting with 1 (for example, Document1, Document2, Document3...).
DocumentCount	The number of documents created in ProcessData.
encoding	The message encoding.
expiry	The message expiry of this message.
feedback	The message feedback of this message.
format	The format of this message.

Parameter	Description
groupId	The group identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
groupStatus	The message group status of this message.
messageFlags	The message flags of this message
messageId	The message identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
messageSequenceNumber	The message sequence number of this message.
messageType	The type of message received.
offset	The current message offset.
persistence	The persistence setting for this message.
priority	The priority setting for this message.
putApplicationName	The put application name of this message.
putApplicationType	The put application type of this message.
putDateTime	The put date and time of this message in the format "MM-dd-yyyy HH:mm:ss"
replyToQueueManagerName	The reply to queue manager of this message.
replyToQueueName	The reply to queue of this message.
report	The contents of the report field of this message.

Examples

Example 1 – Process Data

The values specified for `wsmq_metadata1` and `wsmq_metadata2` determine which metadata fields are included with each document. The following example is what process data would look like as a result of specifying 4095 for both `wsmq_metadata1` and `wsmq_metadata2`:

```
<ProcessData>
  <WSMQ>
    <DocumentCount>1</documentCount>
    <Document1 SCIObjectID="df8f5e:102fa5a6c8f:-7414">
      <messageId>414D5120514D5F6761727931303030205D624C4220000502</messageId>
      <correlationId>0000000000000000000000000000000000000000000000000000000000000000</correlationId>
      <groupId>0000000000000000000000000000000000000000000000000000000000000000</groupId>
    </Document1>
  </WSMQ>
  <accountingToken>16010515000000B5E512BBA14EC03000000000000000B</accountingToken>
  <replyToQueueManagerName>QM_gary1000
</replyToQueueManagerName>
  <replyToQueueName/>
  <applicationIdData/>
```



```

    <applicationOriginData/>
    <messageType verbose="Datagram">8</messageType>
    <format>MQSTR    </format>
    <report>0</report>
    <feedback>0</feedback>
    <groupStatus/>
    <encoding>273</encoding>
    <characterSet>819</characterSet>
    <expiry>-1</expiry>
    <putDateTime>03-31-2005 15:58:18</putDateTime>
    <putApplicationName>MQSeries Client for Java    </putApplicationName>
    <putApplicationType>28</putApplicationType>
    <messageFlags>0</messageFlags>
    <messageSequenceNumber>1</messageSequenceNumber>
    <offset>0</offset>
    <persistence>0</persistence>
    <priority>0</priority>
  </Document1>
</WSMQ>
</ProcessData>

```

Example 2 – Processing Batch Documents Using GETALL

The following is an example of how to process batch documents received from a WebSphereMQ server. The following example could be used as the bootstrap workflow defined in the WebSphereMQ Suite Async Rcv adapter. This example iterates through all batch documents received and extracts each of them using the File System adapter:

```

<process name="(MyBusinessProcess) ">
  <rule name="has more docs">
    <condition>number(/ProcessData/index/text()) &lt;
number(/ProcessData/WSMQ/DocumentCount/text())</condition>
  </rule>

  <sequence>
    <assign to="index">0</assign>
    <choice name="for each doc">
      <select>
        <case ref="has more docs" activity="process doc"/>
      </select>

      <sequence name="process doc">
        <assign to="index" from="number(number(index) + 1)"></assign>
        <operation name="File System Adapter">
          <participant name=" (My_File_System_Adapter)"/>
          <output message="FileSystemInputMessage">
            <assign to="." from="*"></assign>
            <assign to="Action">FS_EXTRACT</assign>
            <assign to="PrimaryDocument"
from="/ProcessData/WSMQ/*[number(/ProcessData/index/text())]/@SCIObjectID"></assign>

          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>

```

```
        <repeat name="next doc" ref="for each doc"/>

        </sequence>
    </choice>
</sequence>
</process>
```

Example 3 – Sending Messages Using the Adapter

To send messages using the WebSphere MQ Suite adapter, set the parameters in a business process, as follows:

```
<PARAM>
<name>SSL_SETTING_ca_cert_ids</name>
<value>Measle1:1decdec:11159ba495b:-583c,frcppe03z3:1037c71:11584abf184:-7c4d,MBradley1:1decdec:11159ba495b:-5837</value>
</PARAM>
<PARAM>
<name>SSL_SETTING_cipherSuite</name>
<value>SSL_RSA_WITH_3DES_EDE_CBC_SHA</value>
</PARAM>
<PARAM>
<name>SSL_SETTING_keyCertID</name>
<value>frcppe03z3:1df073d:1153772f2cb:-66de</value>
</PARAM>
<PARAM>
<name>SSL_SETTING_ssl_option</name>
<value>SSL_MUST</value>
</PARAM>
```

WebSphereMQ Suite Backout Service

The WebSphereMQ Suite Backout service is used in conjunction with the other WebSphereMQ Suite services to send and receive messages from a WebSphereMQ server. The following table provides an overview of the WebSphereMQ Suite Backout service:

System Name	WSMQBackout
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite to backout SyncPoint operations for a given WebSphereMQ Queue Manager.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send and/or receive messages from a WebSphereMQ server.
Usage example	Should only be called if SyncPoint operations need to be backed out.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name isWSMQ_Backout.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQOpenSession, WSMQCloseSession, WSMQOpenQueue, WSMQCloseQueue, WSMQPutMessage, WSMQGetMessage, WSMQCommit
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes or process any business process documents.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters
Business process context considerations	Should only be called if SyncPoint operations need to be backed out.
Returned status values	<ul style="list-style-type: none">◆ Success – Backout was successful.◆ Error – Backout was unsuccessful.
Restrictions	None
Persistence level	Default
Testing considerations	While testing this service, it is recommended to turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.

Implementing the WebSphereMQ Suite Backout Service

There are no steps required for implementing this service. The default configuration, WSMQ_Backout, is ready for use as soon as Application is installed.

Configuring the WebSphereMQ Suite Backout Service

This service has no configurable parameters.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the WebSphereMQ Suite Backout service:

Parameter	Description
wsmq_sessionid	The session identifier to use from a previously established session. Valid value is the value from a previously opened session. Required.

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the WebSphereMQ Suite Backout service to the business process:

Parameter	Description
WSMQ/CompletionCode	Only set if an MQException occurs and indicates the completion code.
WSMQ/ReasonCode	Only set if an MQException occurs and indicates the reason code.

Business Process Example

The following example illustrates how the WebSphereMQ Suite Backout service can be used in a business process:

```
<process name = "PUTexample">
  <sequence name="OpenSession">
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_hostname">some_host_name</assign>
        <assign to="wsmq_channel">some_channel_name</assign>
        <!-- <assign to="wsmq_debug">Yes</assign> -->
      </output>
      <input message="fromService">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```

```

</operation>

<sequence name="SessionOpened">
  <operation name="Open Queue">
    <participant name="WSMQ_OpenQueue"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
      <assign to="wsmq_qname">some_queue_name</assign>
      <assign to="wsmq_MQOO_type">PUT</assign>
      <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
    </output>
    <input message="fromService"/>
  </operation>

  <sequence name="PUTgroup">
    <sequence name="PUTsequence">
      <operation name="PUT message(s)">
        <participant name="WSMQ_PutMessage"/>
        <output message="toService">
          <assign to="." from="*" />
          <assign to="wsmq_qname">some_queue_name</assign>
        </output>
        <input message="fromService">
          <assign to="." from="*" />
        </input>
      </operation>
    </sequence> <!-- end of PUTsequence -->
    <onFault>
      <operation name="Backout">
        <participant name="WSMQ_Backout"/>
        <output message="toService">
          <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
        </output>
        <input message="fromService"/>
      </operation>
    </onFault>
    <operation name="Commit the PUT">
      <participant name="WSMQ_Commit"/>
      <output message="toService">
        <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
      </output>
      <input message="fromService"/>
    </operation>
  </sequence> <!-- end of PUTgroup -->

  <operation name="Close Queue">
    <participant name="WSMQ_CloseQueue"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
      <assign to="wsmq_qname">some_queue_name</assign>
    </output>
    <input message="fromService"/>
  </operation>

  <sequence name="CloseSession">
    <operation name="Close Session">

```

```
<participant name="WSMQ_CloseSession"/>
<output message="toService">
  <assign to="." from="*" /> <!--wsmq_sessionid passed by default -->
</output>
<input message="fromService"/>
</operation>
</sequence>

<onFault>
  <operation name="Make sure session is closed">
    <participant name="WSMQ_CloseSession"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
    </output>
    <input message="fromService"/>
  </operation>
</onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence> <!-- end of OpenSession sequence -->
</process>
```

WebSphereMQ Suite Close Queue Service

The WebSphereMQ Suite Close Queue service is used in conjunction with the other WebSphereMQ Suite services to send and receive messages from a WebSphereMQ server. The following table provides an overview of the WebSphereMQ Suite Close Queue service:

System Name	WSMQCloseQueue
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite to close a queue.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send and/or receive messages from a WebSphereMQ server
Usage example	Should be called after all puts and/or gets have been performed to a given queue.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name is WSMQ_CloseQueue
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQOpenSession, WSMQCloseSession, WSMQOpenQueue, WSMQPutMessage, WSMQGetMessage, WSMQCommit, WSMQBackout
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes or process any business process documents.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters
Business process context considerations	This service is not mandatory but is recommended to ensure expected results. The Close Session service will automatically close any queues still open. Furthermore, the Queue Manager will close any handles still opened once disconnected.
Returned status values	<ul style="list-style-type: none">◆ Success – Queue closed successfully.◆ Error – This service never sets an error status.
Restrictions	None
Persistence level	Default
Testing considerations	While testing this service, it is recommended to turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.

Implementing the WebSphereMQ Suite Close Queue Service

There are no steps required for implementing this service. The default configuration, WSMQ_CloseQueue, is ready for use as soon as Application is installed. However, you can edit the service's parameter settings for specific business processes in the GPM.

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
Queue Name (wsmq_qname)	The name of the queue to close. Required. Valid value is alphanumeric.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the WebSphereMQ Suite Close Queue service:

Parameter	Description
Queue Name (wsmq_qname)	The name of the queue to close. Required. Valid value is alphanumeric.
wsmq_sessionid	The session identifier to use from a previously established session. Valid value is the value from a previously opened session. Required.

Business Process Example

The following example illustrates how the WebSphereMQ Suite Close Queue service can be used in a business process:

```
<process name = "PUTexample">
  <sequence name="OpenSession">
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_hostname">some_host_name</assign>
        <assign to="wsmq_channel">some_channel_name</assign>
        <!-- <assign to="wsmq_debug">Yes</assign> -->
      </output>
      <input message="fromService">
        <assign to="." from="*" />
      </input>
    </operation>
  </sequence>
</process>
```



```

<sequence name="SessionOpened">
  <operation name="Open Queue">
    <participant name="WSMQ_OpenQueue"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
      <assign to="wsmq_qname">some_queue_name</assign>
      <assign to="wsmq_MQOO_type">PUT</assign>
      <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
    </output>
    <input message="fromService"/>
  </operation>

  <operation name="PUT message(s)">
    <participant name="WSMQ_PutMessage"/>
    <output message="toService">
      <assign to="." from="*" />
      <assign to="wsmq_qname">some_queue_name</assign>
    </output>
    <input message="fromService">
      <assign to="." from="*" />
    </input>
  </operation>

  <operation name="Commit the PUT">
    <participant name="WSMQ_Commit"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
    </output>
    <input message="fromService"/>
  </operation>

  <operation name="Close Queue">
    <participant name="WSMQ_CloseQueue"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
      <assign to="wsmq_qname">some_queue_name</assign>
    </output>
    <input message="fromService"/>
  </operation>

  <sequence name="CloseSession">
    <operation name="Close Session">
      <participant name="WSMQ_CloseSession"/>
      <output message="toService">
        <assign to="." from="*" /> <!--wsmq_sessionid passed by default -->
      </output>
      <input message="fromService"/>
    </operation>
  </sequence>

  <onFault>
    <operation name="Make sure session is closed">
      <participant name="WSMQ_CloseSession"/>
      <output message="toService">
        <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
      </output>
    </operation>
  </onFault>

```

```
        <input message="fromService"/>
    </operation>
</onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence> <!-- end of OpenSession sequence -->
</process>
```

WebSphereMQ Suite Close Session Service

The WebSphereMQ Suite Close Session service is used in conjunction with the other WebSphereMQ Suite services to send and receive messages from a WebSphereMQ server. The following table provides an overview of the WebSphereMQ Suite Close Session service:

System Name	WSMQCloseSession
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite to close an existing session and disconnect from the WebSphereMQ Queue Manager.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send and/or receive messages from a WebSphereMQ server.
Usage example	Should be called last after all other operations have been performed.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name is WSMQ_CloseSession.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQOpenSession, WSMQOpenQueue, WSMQCloseQueue, WSMQPutMessage, WSMQGetMessage, WSMQCommit, WSMQBackout
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes or process any business process documents.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters
Business process context considerations	If a session is opened but not closed, connections to the queue manager will remain open. Periodic checking for stale sessions is performed but it is still recommended that this service be placed in an OnFault handler (see bpml example below).
Returned status values	<ul style="list-style-type: none">◆ Success – Session closed and disconnected from the Queue Manager.◆ Error – This service never sets an error status.
Restrictions	None
Persistence level	Default
Testing considerations	While testing this service, it is recommended to turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.

Implementing the WebSphereMQ Suite Close Session Service

There are no steps required for implementing this service. The default configuration, WSMQ_CloseSession, is ready for use as soon as Application is installed.

Configuring the WebSphereMQ Suite Close Session Service

This service has no configurable parameters.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the WebSphereMQ Suite Close Session service:

Parameter	Description
wsmq_sessionid	The session identifier to use from a previously established session. Valid value is the value from a previously opened session. Required.

Business Process Example

The following example illustrates how the WebSphereMQ Suite Close Session service can be used in a business process:

```
<process name = "PUTexample">
  <sequence name="OpenSession">
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_hostname">some_host_name</assign>
        <assign to="wsmq_channel">some_channel_name</assign>
        <!-- <assign to="wsmq_debug">Yes</assign> -->
      </output>
      <input message="fromService">
        <assign to="." from="*" />
      </input>
    </operation>

    <sequence name="SessionOpened">
      <operation name="Open Queue">
        <participant name="WSMQ_OpenQueue"/>
        <output message="toService">
          <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
          <assign to="wsmq_qname">some_queue_name</assign>
          <assign to="wsmq_MQOO_type">PUT</assign>
          <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
        </output>
        <input message="fromService" />
      </operation>
    </sequence>
  </sequence>
</process>
```

```

    <operation name="PUT message(s)">
      <participant name="WSMQ_PutMessage"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_qname">some_queue_name</assign>
<!-- If neither two following assign's are specified, the PrimaryDocument will be
sent
OR to perform batch sending, use the following assign:
      <assign to="wsmq_batchFilter">*</assign>
OR to send some other single document, use the following assign:
      <assign to="wsmq_document" from="SomeDocumentName/@*" />
-->
    </output>
    <input message="fromService">
      <assign to="." from="*" />
    </input>
  </operation>

  <operation name="Commit the PUT">
    <participant name="WSMQ_Commit"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
    </output>
    <input message="fromService"/>
  </operation>

  <operation name="Close Queue">
    <participant name="WSMQ_CloseQueue"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
      <assign to="wsmq_qname">some_queue_name</assign>
    </output>
    <input message="fromService"/>
  </operation>

  <sequence name="CloseSession">
    <operation name="Close Session">
      <participant name="WSMQ_CloseSession"/>
      <output message="toService">
        <assign to="." from="*" /> <!--wsmq_sessionid passed by default -->
      </output>
      <input message="fromService"/>
    </operation>
  </sequence>

  <onFault>
    <operation name="Make sure session is closed">
      <participant name="WSMQ_CloseSession"/>
      <output message="toService">
        <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
      </output>
      <input message="fromService"/>
    </operation>
  </onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence> <!-- end of OpenSession sequence -->

```

</process>

WebSphereMQ Suite Commit Service

The WebSphereMQ Suite Commit service is used in conjunction with the other WebSphereMQ Suite services to send and receive messages from a WebSphereMQ server. The following table provides an overview of the WebSphereMQ Suite Commit service:

System Name	WSMQCommit
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite to commit SyncPoint operations for a given WebSphereMQ Queue Manager.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send and/or receive messages from a WebSphereMQ server.
Usage example	Should be called after SyncPoint put and/or get operations have been performed.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name is WSMQ_Commit.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQOpenSession, WSMQCloseSession, WSMQOpenQueue, WSMQCloseQueue, WSMQPutMessage, WSMQGetMessage, WSMQBackout
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes or process any business process documents.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters
Business process context considerations	<p>This service is not necessary at all if SyncPoint operations are not performed. And even though this service is not mandatory even with SyncPoint operations, because the QueueManager (on some platforms) will perform the commits automatically, its use is recommended to ensure expected results.</p> <p>WSMQCommit has no bearing if it exists within a distributed transaction. It is ignored. The commit is done on the transaction end, so this service is not required (or used) within the business process.</p>
Returned status values	<ul style="list-style-type: none">◆ Success – Commit was successful.◆ Error – Commit was unsuccessful.
Restrictions	WSMQCommit has no bearing if it exists within a distributed transaction. It is ignored. The commit is done on the transaction end, so this service is not required (or used) within the business process.
Persistence level	Default

Testing considerations	While testing this service, it is recommended to turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.
------------------------	---

Implementing the WebSphereMQ Suite Commit Service

There are no steps required for implementing this service. The default configuration, WSMQ_Commit, is ready for use as soon as Application is installed.

Configuring the WebSphereMQ Suite Commit Service

This service has no configurable parameters.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the WebSphereMQ Suite Commit service:

Parameter	Description
wsmq_sessionid	The session identifier to use from a previously established session. Valid value is the value from a previously opened session. Required.

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the WebSphereMQ Suite Commit service to the business process:

Parameter	Description
WSMQ/CompletionCode	Only set if an MQException occurs and indicates the completion code.
WSMQ/ReasonCode	Only set if an MQException occurs and indicates the reason code.

Business Process Example

The following example illustrates how the WebSphereMQ Suite Commit service can be used in a business process:

```
<process name = "PUTexample">
  <sequence name="OpenSession">
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_hostname">some_host_name</assign>
        <assign to="wsmq_channel">some_channel_name</assign>
      </output message>
    </operation>
  </sequence>
</process>
```



```

    <!-- <assign to="wsmq_debug">Yes</assign> -->
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

<sequence name="SessionOpened">
  <operation name="Open Queue">
    <participant name="WSMQ_OpenQueue" />
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
      <assign to="wsmq_qname">some_queue_name</assign>
      <assign to="wsmq_MQOO_type">PUT</assign>
      <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
    </output>
    <input message="fromService" />
  </operation>

  <operation name="PUT message(s)">
    <participant name="WSMQ_PutMessage" />
    <output message="toService">
      <assign to="." from="*" />
      <assign to="wsmq_qname">some_queue_name</assign>
    </output>
    <input message="fromService" />
  </operation>

  <!-- If neither two following assign's are specified, the PrimaryDocument will be
  sent
  OR to perform batch sending, use the following assign:
  <assign to="wsmq_batchFilter">*</assign>
  OR to send some other single document, use the following assign:
  <assign to="wsmq_document" from="SomeDocumentName/@*" />
  -->
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

<operation name="Commit the PUT">
  <participant name="WSMQ_Commit" />
  <output message="toService">
    <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
  </output>
  <input message="fromService" />
</operation>

<operation name="Close Queue">
  <participant name="WSMQ_CloseQueue" />
  <output message="toService">
    <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
    <assign to="wsmq_qname">some_queue_name</assign>
  </output>
  <input message="fromService" />
</operation>

<sequence name="CloseSession">
  <operation name="Close Session">

```

```
<participant name="WSMQ_CloseSession"/>
<output message="toService">
  <assign to="." from="*" /> <!--wsmq_sessionid passed by default -->
</output>
<input message="fromService"/>
</operation>
</sequence>

<onFault>
  <operation name="Make sure session is closed">
    <participant name="WSMQ_CloseSession"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
    </output>
    <input message="fromService"/>
  </operation>
</onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence> <!-- end of OpenSession sequence -->
</process>
```

WebSphereMQ Suite Get Message Service

The WebSphereMQ Suite Get Message service is used as part of the WebSphereMQ Suite to open a queue. The following table provides an overview of the WebSphereMQ Suite Get Message service:

System Name	WSMQGetMessage
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite to get or browse messages in a queue.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send and/or receive messages from a WebSphereMQ server.
Usage example	Used to “get” messages off a queue.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name is WSMQ_GetMessage.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQOpenSession, WSMQCloseSession, WSMQOpenQueue, WSMQCloseQueue, WSMQPutMessage, WSMQCommit, WSMQBackout
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes. All data received is placed into process data.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters.
Business process context considerations	Documents created by this service as a result of receiving messages are placed directly into process data. Use wsmq_metadata1 and/or wsmq_metadata2 to specify what metadata is included with the document from the message(s) that comprise the document.
Returned status values	<ul style="list-style-type: none">◆ Success – GetMessage service completed successfully.◆ Error – GetMessage service did not complete successfully.
Restrictions	None
Persistence level	Default
Testing considerations	While testing this service, it is recommended to turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.

Implementing the WebSphereMQ Suite Get Message Service

There are no steps required for implementing this service. The default configuration, WSMQ_GetMessage, is ready for use as soon as Application is installed. However, you can edit the service's parameter settings for specific business processes in the GPM.

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
Browse Lock Options (wsmq_browselock)	Only valid when browsing messages. Optional. Valid values are: <ul style="list-style-type: none">◆ Lock message (to use in code, specify LOCK)◆ Unlock message ((to use in code, specify UNLOCK)
Buffersize override (wsmq_buffersize)	If specified, overrides the default buffersize used when streaming data. Valid values are 0-999999999. Optional.
Document Name (wsmq_docName)	The document name to associate with the data received. Default is %^.dat. Optional.
Document tracking (wsmq_docTracking)	Specifies whether to perform document tracking. Valid values are Yes and No. Default is No. Optional.
Error Status For MQException (wsmq_errorOnMQException)	Normally when an MQException occurs, the workflow status is set to error but this allows it to be set to success so that the workflow can perform its own error process or retry logic. Valid values are Yes and No. Default is Yes. Optional.
Group messages (wsmq_groupBy)	Selects which identifier to use when grouping messages. Only used if the wsmq_msgHandling parm is set to GROUP. Optional. Valid values are: <ul style="list-style-type: none">◆ With the same group identifier (GRPID)◆ With the same message identifier (MSGID)◆ With the same correlation identifier (CORID)
Group messages even when the identifier is MQ*_NONE? (wsmq_groupMsgWhenIdNone)	Determines if messages will be grouped even if the identifier equals MQ*_NONE. Valid values are Yes and No. Default is Yes. Optional.
Convert wsmq_MQMO_corld from hex to byte array (wsmq_hexToByte_corld)	Select Yes to invoke the conversion from hex to byte array. Valid values are Yes and No. Default is No. Optional.
Convert wsmq_MQMO_grpld from hex to byte array (wsmq_hexToByte_grpld)	Select Yes to invoke the conversion from hex to byte array. Valid values are Yes and No. Default is No. Optional.

Field	Description
Convert wsmq_MQMO_msgId from hex to byte array (wsmq_hexToByte_msgId)	Select Yes to invoke the conversion from hex to byte array. Valid values are Yes and No. Default is No. Optional.
Metadata1 To Include (wsmq_metadata1)	<p>Specifies which metadata fields (add values together for multiple fields) from the message to include with the document created in process data. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 0 – no metadata fields (default) ◆ 1 – msgId ◆ 2 – corId ◆ 4 – grpId ◆ 8 – msgType ◆ 16 – replyToQM ◆ 32 – replyToQ ◆ 64 – acctToken ◆ 128 – ApplIdData ◆ 256 – ApplOrigin ◆ 512 – Format ◆ 1024 – Report ◆ 2048 – Feedback
Metadata2 To Include (wsmq_metadata2)	<p>Specifies which metadata fields (add values together for multiple fields) from the message to include with the document created in process data. Optional. Valid values are:</p> <ul style="list-style-type: none"> ◆ 0 – no metadata fields (default) ◆ 1 – groupStatus ◆ 2 – encoding ◆ 4 – charset ◆ 8 – expiry ◆ 16 – putDateTime ◆ 32 – putAppName ◆ 64 – putAppType ◆ 128 – msgFlags ◆ 256 – msgSeqNum ◆ 512 – offset ◆ 1024 – persistence ◆ 2048 – priority

Field	Description
MQGMO_ALL_MSGS_AVAILABLE (wsmq_MQGMO_allmsgavail)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is Yes. Optional.
MQGMO_ALL_SEGMENTS_AVAILABLE (wsmq_MQGMO_allsegavail)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is Yes. Optional.
MQGMO_COMPLETE_MSG (wsmq_MQGMO_completemsg)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is Yes. Optional.
MQGMO_CONVERT (wsmq_MQGMO_convert)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is No. Optional.
MQGMO_FAIL_IF QUIESCING (wsmq_MQGMO_failifquiescing)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is Yes. Optional.
MQGMO_LOGICAL_ORDER (wsmq_MQGMO_logicalorder)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is Yes. Optional.
MQGMO_SYNCPOINT (wsmq_MQGMO_syncpoint)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is Yes. Optional. Note: When using the BROWSE functions for Get Type (wsmq_type), the MQGMO_SYNCPOINT (wsmq_MQGCO_syncpoint) parameter must be set to No.
MQGMO_WAIT (wsmq_MQGMO_wait)	Select Yes to include this Get Message Option. Valid values are Yes and No. Default is No. Optional.
Wait Interval (milliseconds) (wsmq_MQGMO_waitInterval)	The wait interval specified in milliseconds. Must select Yes on wsmq_MQGMO_wait for this value to be used. Valid values are 1-999999999. Optional.
MQMO_MATCH_CORREL_ID (wsmq_MQMO_corId)	Sets the MQMO_MATCH_CORREL_ID option to this value. Optional.
MQMO_MATCH_GROUP_ID (wsmq_MQMO_grpld)	Sets the MQMO_MATCH_GROUP_ID option to this value. Optional.
MQMO_MATCH_MSG_ID (wsmq_MQMO_msgId)	Sets the MQMO_MATCH_MSG_ID. Optional.
Message Handling (wsmq_msgHandling)	Select the type of message handling to be used. Optional. Valid values are: <ul style="list-style-type: none"> ◆ Receive one message as one document (ONE) (default) ◆ Receive all messages in a logical group as one document (GROUP) ◆ Receive all messages available as one document (ALL) ◆ Combine all messages with the same msgId and trigger using msgType (SPEC1)
Queue Name (wsmq_qname)	The name of a previously opened queue used to PUT messages. Required.

Field	Description
Receive Message Limit (wsmq_rcvMsgLimit)	Used with GETALL or BROWSEALL to limit the number of messages received. Valid values are 0-999999999. Default is 0 (unlimited). Optional.
Use the group status flag to determine end-of-group? (wsmq_useGroupStatus)	Specifies whether to use the group status to determine end of a group or to receive until no more messages are available. Valid values are Yes and No. Default is No. Optional.
Get Type (wsmq_type)	<p>Specifies the type of Get to perform. These types are better defined below. Optional. Valid values are (the following are codes only):</p> <ul style="list-style-type: none"> ◆ GETONE – Destructively gets the first message in the queue that satisfies the selection criteria. (default) ◆ GETALL – Destructively gets all messages in the queue that satisfy the selection criteria. ◆ GETUNDERCURSOR – Destructively gets the message pointed to by the browser cursor regardless of the selection criteria. ◆ BROWSEALL – Non-destructively gets all messages in the queue that satisfy the selection criteria (cannot use lock). The following BPML is a sample of how you could use BROWSEALL: <pre> <operation name="GET message(s)"> <participant name="WSMQ_GetMessage"/> <output message="toService"> <assign to="wsmq_metadata1">4095</assign> <assign to="wsmq_metadata2">4095</assign> <assign to="wsmq_qname">qname</assign> <assign to="wsmq_type">BROWSEALL</assign> <assign to="wsmq_MQGMO_syncpoint">No</assign> <assign to="wsmq_rcvMsgLimit">3</assign> <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"></assign> </output> <input message="fromService"> <assign to="." from="*"></assign> </input> </operation> </pre> <ul style="list-style-type: none"> ◆ BROWSENEXT – Non-destructively gets the next message in the queue that satisfies the selection criteria. ◆ BROWSEFIRST – Non-destructively gets the first message in the queue that satisfies the selection criteria. ◆ BROWSEUNDERCURSOR – Non-destructively gets the message currently under the browse cursor regardless of the selection criteria. <p>Note: When using the BROWSE functions for Get Type (wsmq_type), the MQGMO_SYNCPOINT (wsmq_MQGCO_syncpoint) parameter must be set to No.</p>

Process Data Example

The following example illustrates how process data would look if you specified the value 4095 for the parameters wsmq_metadata1 and wsmq_metadata2. (4095 is the sum of all the available values for the metadata fields: 1+2+4+8+16+32+64+128+256+512+1024+2048 = 4095):

```
<ProcessData>
  <WSMQ>
    <DocumentCount>1</documentCount>
    <Document1 SCIObjectID="df8f5e:102fa5a6c8f:-7414">
      <messageId>414D5120514D5F6761727931303030205D624C4220000502</messageId>
      <correlationId>0000000000000000000000000000000000000000000000000000000000000000</correlationId>
      <groupId>0000000000000000000000000000000000000000000000000000000000000000</groupId>

    <accountingToken>16010515000000B5E512BBA14EC030000000000000000B</accountingToken>
      <replyToQueueManagerName>QM_test1000
    </replyToQueueManagerName>
      <replyToQueueName/>
      <applicationIdData/>
      <applicationOriginData/>
      <messageType verbose="Datagram">8</messageType>
      <format>MQSTR    </format>
      <report>0</report>
      <feedback>0</feedback>
      <groupStatus/>
      <encoding>273</encoding>
      <characterSet>819</characterSet>
      <expiry>-1</expiry>
      <putDateTime>03-31-2005 15:58:18</putDateTime>
      <putApplicationName>MQSeries Client for Java    </putApplicationName>
      <putApplicationType>28</putApplicationType>
      <messageFlags>0</messageFlags>
      <messageSequenceNumber>1</messageSequenceNumber>
      <offset>0</offset>
      <persistence>0</persistence>
      <priority>0</priority>
    </Document1>
  </WSMQ>
</ProcessData>
```

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the WebSphereMQ Suite Get Message service to the business process:

Parameter	Description
accountingToken	The accounting token of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
applicationIdData	The application identifier data of this message
applicationOriginData	The application origin data of this message

Parameter	Description
characterSet	The character set of this message
correlationId	The correlation identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
DocumentX	Documents received using this service are always created sequentially starting with "1" (for example, Document1, Document2, Document3).
DocumentCount	The number of documents created in ProcessData.
encoding	The message encoding.
expiry	The message expiry of this message.
feedback	The message feedback of this message.
format	The format of this message.
groupId	The group identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
groupStatus	The message group status of this message.
messageFlags	The message flags of this message
messageId	The message identifier of this message. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
messageSequenceNumber	The message sequence number of this message.
messageType	The type of message received.
offset	The current message offset
persistence	The persistence setting for this message.
priority	The priority setting for this message.
putApplicationName	The put application name of this message.
putApplicationType	The put application type of this message.
putDateTime	The put date and time of this message in the format "MM-dd-yyyy HH:mm:ss"
replyToQueueManagerName	The reply to queue manager of this message.
replyToQueueName	The reply to queue of this message.
report	The contents of the report field of this message.
WSMQ/CompletionCode	Only set if an MQException occurs and indicates the completion code.
WSMQ/ReasonCode	Only set if an MQException occurs and indicates the reason code.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the WebSphereMQ Suite Get Message service:

Parameter	Description
All GPM parameters	See <i>Setting Up the Service in the GPM</i> .
wsmq_sessionid	The session identifier to use from a previously established session. Valid value is the value from a previously opened session. Required.

Business Process Example

The following example illustrates how the WebSphereMQ Suite Get Message service can be used in a business process:

```
<process name = "GETexample">
  <sequence name="OpenSession">
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_hostname">some_host_name</assign>
        <assign to="wsmq_channel">some_channel_name</assign>
        <!-- <assign to="wsmq_debug">Yes</assign> -->
      </output>
      <input message="fromService">
        <assign to="." from="*" />
      </input>
    </operation>

    <sequence name="SessionOpened">
      <operation name="Open Queue">
        <participant name="WSMQ_OpenQueue"/>
        <output message="toService">
          <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
          <assign to="wsmq_qname">some_queue_name</assign>
          <assign to="wsmq_MQOO_type">PUT</assign>
          <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
        </output>
        <input message="fromService"/>
      </operation>

      <operation name="GET message(s)">
        <participant name="WSMQ_GetMessage"/>
        <output message="toService">
          <assign to="wsmq_sessionid" from="string(wsmq_sessionid)"/>
          <assign to="wsmq_qname" from="some_queue_name"/>
          <assign to="wsmq_metadata1">4095</assign>
          <assign to="wsmq_metadata2">4095</assign>
          <assign to="wsmq_type">GETALL</assign>
        </output>
        <input message="fromService">
```

```

    <assign to="." from="*" />
  </input>
</operation>

<operation name="Commit the GET">
  <participant name="WSMQ_Commit" />
  <output message="toService">
    <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
  </output>
  <input message="fromService" />
</operation>

<operation name="Close Queue">
  <participant name="WSMQ_CloseQueue" />
  <output message="toService">
    <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
    <assign to="wsmq_qname">some_queue_name</assign>
  </output>
  <input message="fromService" />
</operation>

<sequence name="CloseSession">
  <operation name="Close Session">
    <participant name="WSMQ_CloseSession" />
    <output message="toService">
      <assign to="." from="*" /> <!--wsmq_sessionid passed by default -->
    </output>
    <input message="fromService" />
  </operation>
</sequence>

<onFault>
  <operation name="Make sure session is closed">
    <participant name="WSMQ_CloseSession" />
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
    </output>
    <input message="fromService" />
  </operation>
</onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence> <!-- end of OpenSession sequence -->
</process>

```

WebSphereMQ Suite Open Queue Service

The WebSphereMQ Suite Open Queue service is used as part of the WebSphereMQ Suite to open a queue. The following table provides an overview of the WebSphereMQ Suite Open Queue service:

System Name	WSMQOpenQueue
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite to open a queue.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send and/or receive messages from a WebSphereMQ server.
Usage example	Must be called before any puts or gets can be performed.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name is WSMQ_OpenQueue.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQOpenSession, WSMQCloseSession, WSMQCloseQueue, WSMQPutMessage, WSMQGetMessage, WSMQCommit, WSMQBackout
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes or process any business process documents.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters.
Business process context considerations	Must be called before any puts or gets can be performed.
Returned status values	<ul style="list-style-type: none">◆ Success – Queue opened successfully.◆ Error – Queue failed to open; check configuration options.
Restrictions	None
Persistence level	Default
Testing considerations	While testing this service, it is recommended to turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.

Implementing the WebSphereMQ Suite Open Queue Service

There are no steps required for implementing this service. The default configuration, WSMQ_OpenQueue, is ready for use as soon as Application is installed. However, you can edit the service's parameter settings for specific business processes in the GPM.

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
Error Status For MQException (wsmq_errorOnMQException)	Normally when an MQException occurs, the workflow status is set to error but this allows it to be set to success so that the workflow can perform its own error process or retry logic. Optional. Valid values are Yes and No. Default is Yes.
Binding Options (wsmq_MQOO_binding)	These options apply when the queue being opened is a cluster queue. Optional. Valid values are: <ul style="list-style-type: none">◆ MQOO_BIND_AS_Q_DEF (ASQDEF) (default)◆ MQOO_BIND_ON_OPEN (ONOPEN)◆ MQOO_BIND_NOT_FIXED (NOTFIXED)
Context Options (wsmq_MQOO_context)	These options control the processing of message context. Optional. Valid values are: <ul style="list-style-type: none">◆ No context setting (NONE) (default)◆ MQOO_SET_IDENTITY_CONTEXT (IDCTX)◆ MQOO_SET_ALL_CONTEXT (ALLCTX)
MQOO_FAIL_IF_QUIESCING? (wsmq_MQOO_failifquiescing)	Indicates whether or not to include the MQOO_FAIL_IF_QUIESCING Open Option. Optional. Valid values are Yes and No. Default is Yes.
Queue open options (wsmq_type)	Specifies how the queue should be opened and for what type of operations will be performed. Required. No default. Valid values are: <ul style="list-style-type: none">◆ Open queue to put messages (PUT)◆ Open queue to get messages (GET)◆ Open queue to browse and/or get messages (BROWSE)◆ Open queue to put and get messages (PUTGET)◆ Open queue to put, get or browse messages (PUTBROWSE)
Queue Name (wsmq_qname)	The name of the queue to open. Required. Valid value is alphanumeric. No default.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the WebSphereMQ Suite Open Queue service:

Parameter	Description
All GPM parameters	See <i>Setting Up the Service in the GPM</i> .
wsmq_sessionid	The session identifier to use from a previously established session. Valid value is the value from a previously opened session. Required.

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the WebSphereMQ Suite Open Queue service to the business process:

Parameter	Description
WSMQ/CompletionCode	Only set if an MQException occurs and indicates the completion code.
WSMQ/ReasonCode	Only set if an MQException occurs and indicates the reason code.

Business Process Example

The following example illustrates how the WebSphereMQ Suite Open Queue service can be used in a business process:

```
<process name = "PUTexample">
  <sequence name="OpenSession">
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_hostname">some_host_name</assign>
        <assign to="wsmq_channel">some_channel_name</assign>
        <!-- <assign to="wsmq_debug">Yes</assign> -->
      </output>
      <input message="fromService">
        <assign to="." from="*" />
      </input>
    </operation>

    <sequence name="SessionOpened">
      <operation name="Open Queue">
        <participant name="WSMQ_OpenQueue"/>
        <output message="toService">
          <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
          <assign to="wsmq_qname">some_queue_name</assign>
          <assign to="wsmq_MQOO_type">PUT</assign>
          <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
        </output>
      </operation>
    </sequence>
  </sequence>
</process>
```

```

    </output>
    <input message="fromService" />
</operation>

<operation name="PUT message(s)">
  <participant name="WSMQ_PutMessage" />
  <output message="toService">
    <assign to="." from="*" />
    <assign to="wsmq_qname">some_queue_name</assign>
  </output>
  <input message="fromService">
    <assign to="." from="*" />
  </input>
</operation>

<operation name="Commit the PUT">
  <participant name="WSMQ_Commit" />
  <output message="toService">
    <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
  </output>
  <input message="fromService" />
</operation>

<operation name="Close Queue">
  <participant name="WSMQ_CloseQueue" />
  <output message="toService">
    <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
    <assign to="wsmq_qname">some_queue_name</assign>
  </output>
  <input message="fromService" />
</operation>

<sequence name="CloseSession">
  <operation name="Close Session">
    <participant name="WSMQ_CloseSession" />
    <output message="toService">
      <assign to="." from="*" /> <!--wsmq_sessionid passed by default -->
    </output>
    <input message="fromService" />
  </operation>
</sequence>

<onFault>
  <operation name="Make sure session is closed">
    <participant name="WSMQ_CloseSession" />
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
    </output>
    <input message="fromService" />
  </operation>
</onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence> <!-- end of OpenSession sequence -->
</process>

```

WebSphere MQ Suite Open Session Service

The WebSphere MQ Suite Open Session service is used in conjunction with the other WebSphereMQ Suite services to send and receive messages from a WebSphereMQ server. The following table provides an overview of the WebSphere MQ Suite Open Session service:

System Name	WSMQOpenSession
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite of services to open a session, which establishes a connection to a WebSphereMQ Queue Manager.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send and/or receive messages from a WebSphereMQ server.
Usage example	Must be called first before any other WebSphereMQ Suite services.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name is WSMQ_OpenSession.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQCloseSession, WSMQOpenQueue, WSMQCloseQueue, WSMQPutMessage, WSMQGetMessage, WSMQCommit, WSMQBackout
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes or process any business process documents.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters
Business process context considerations	If multiple connections (sessions) to different queue managers is required within the same business process, it is the responsibility of the business process writer to save off the wsmq_sessionid values returned from this service and pass them into the appropriate services where needed.
Returned status values	<ul style="list-style-type: none">◆ Success – Session started and connected to the Queue Manager.◆ Error – Session failed to start, usually because it could not connect.
Restrictions	None
Persistence level	Default
Testing considerations	Debug messages can be turned on as part of the business process without having to turn on full system debugging. All logging for this suite of services goes to a separate log file called WebSphereMQSuite. MQ tracing can also be activated as part of the service configuration.

Implementing the WebSphere MQ Suite Open Session Service

There are no steps required for implementing this service. The default configuration, WSMQ_OpenSession, is ready for use as soon as Application is installed. However, you can edit the service's parameter settings for specific business processes in the GPM.

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
Channel (wsmq_channel)	The channel to use. Valid value is alphanumeric. No default. Required.
Turn on debug messages? (wsmq_debug)	Used to turn on debugging messages for this session. Valid values are Yes and No. Default is No. Optional.
Error Status For MQException (wsmq_errorOnMQException)	Normally when an MQException occurs, the workflow status is set to error but this allows it to be set to success so that the workflow can perform its own error process or retry logic. Valid values are Yes and No. Default is Yes. Optional.
Host Name (wsmq_hostname)	The host name or IP address of the WebSphereMQ server. Valid value is alphanumeric. Optional; if not specified, bindings mode is used.
Password (wsmq_password)	A user password if required to access the WebSphereMQ server. Valid value is alphanumeric. No default. Optional.

Field	Description																																						
CCSID (mq_ccsid)	<p>If needed, enter the Coded Character Set Identifier (CCSID) that represents the codeset name you wish to use. Optional.</p> <table border="1"> <thead> <tr> <th>Codeset name</th> <th>Use CCSID</th> </tr> </thead> <tbody> <tr><td>ISO 8859-1</td><td>819 (default)</td></tr> <tr><td>ISO 8859-2</td><td>912</td></tr> <tr><td>ISO 8859-3</td><td>913</td></tr> <tr><td>ISO 8859-5</td><td>915</td></tr> <tr><td>ISO 8859-6</td><td>1089</td></tr> <tr><td>ISO 8859-7</td><td>813</td></tr> <tr><td>ISO 8859-8</td><td>916</td></tr> <tr><td>ISO 8859-9</td><td>920</td></tr> <tr><td>ISO 8859-13</td><td>921</td></tr> <tr><td>ISO 8859-15</td><td>923</td></tr> <tr><td>big5</td><td>950</td></tr> <tr><td>eucJP</td><td>954, 5050, 33722</td></tr> <tr><td>eucKR</td><td>970</td></tr> <tr><td>eucTW</td><td>964</td></tr> <tr><td>eucCN</td><td>1383</td></tr> <tr><td>PCK</td><td>943</td></tr> <tr><td>GBK</td><td>1386</td></tr> <tr><td>koi8-r</td><td>878</td></tr> </tbody> </table> <p>Note: CCSID is not used when connecting directly using "binding moce."</p>	Codeset name	Use CCSID	ISO 8859-1	819 (default)	ISO 8859-2	912	ISO 8859-3	913	ISO 8859-5	915	ISO 8859-6	1089	ISO 8859-7	813	ISO 8859-8	916	ISO 8859-9	920	ISO 8859-13	921	ISO 8859-15	923	big5	950	eucJP	954, 5050, 33722	eucKR	970	eucTW	964	eucCN	1383	PCK	943	GBK	1386	koi8-r	878
Codeset name	Use CCSID																																						
ISO 8859-1	819 (default)																																						
ISO 8859-2	912																																						
ISO 8859-3	913																																						
ISO 8859-5	915																																						
ISO 8859-6	1089																																						
ISO 8859-7	813																																						
ISO 8859-8	916																																						
ISO 8859-9	920																																						
ISO 8859-13	921																																						
ISO 8859-15	923																																						
big5	950																																						
eucJP	954, 5050, 33722																																						
eucKR	970																																						
eucTW	964																																						
eucCN	1383																																						
PCK	943																																						
GBK	1386																																						
koi8-r	878																																						
Listening Port (wsmq_port)	The listening port of the WebSphereMQ server. Valid value is valid port. Default is 1414. Optional.																																						
Queue Manager (wsmq_qmanager)	The Queue Manager name to use. Valid value is alphanumeric. Optional; if not specified, uses the default queue manager																																						
Session Timeout Minutes (wsmq_sessiontimeout)	Overrides the default session timeout. Valid value is numeric, specified in minutes. Default is 15 (minutes). Optional.																																						
Turn on MQ tracing? (wsmq_trace)	Used to turn on MQ level 5 tracing. Instance must be cycled (disable/enabled) to take affect. Valid values are Yes and No. Default is No. Optional.																																						
Trace output filename (wsmq_traceFile)	User specified fully qualified path and filename to use for trace output. Any value is valid. No default. Optional.																																						
Trace Level (1-5) (wsmq_traceLevel)	User specified fully qualified path and filename to use for trace output. Any value is valid. No default. Optional.																																						
User Identifier (wsmq_userid)	A user identifier if required to access the WebSphereMQ server. Valid value is alphanumeric. No default. Optional.																																						

Parameters Passed From Business Process to Service

The parameters passed from the business process to the WebSphere MQ Suite Open Session service are the same as those configurable in the GPM, listed in *Setting Up the Service in the GPM*.

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the WebSphere MQ Suite Open Session service to the business process:

Parameter	Description
wsmq_sessionid	The session identifier is a Global Unique Identifier (GUID) that is then used by all other WebSphereMQ Suite services as it maintains the connection to a queue manager for subsequent operations.
WSMQ/CompletionCode	Only set if an MQException occurs and indicates the completion code.
WSMQ/ReasonCode	Only set if an MQException occurs and indicates the reason code.

Business Process Example

The following example illustrates how the WebSphere MQ Suite Open Session service can be used in a business process:

```
<process name = "PUTexample">
  <sequence name="OpenSession">
    <operation name="Open Session">
      <participant name="WSMQ_OpenSession"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_hostname">some_host_name</assign>
        <assign to="wsmq_channel">some_channel_name</assign>
        <!-- <assign to="wsmq_debug">Yes</assign> -->
      </output>
      <input message="fromService">
        <assign to="." from="*" />
      </input>
    </operation>

    <sequence name="SessionOpened">
      <operation name="Open Queue">
        <participant name="WSMQ_OpenQueue"/>
        <output message="toService">
          <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
          <assign to="wsmq_qname">some_queue_name</assign>
          <assign to="wsmq_MQOO_type">PUT</assign>
          <assign to="wsmq_MQOO_failifquiescing">Yes</assign>
        </output>
        <input message="fromService" />
      </operation>
    </sequence>
  </sequence>
</process>
```

```

    <operation name="PUT message(s)">
      <participant name="WSMQ_PutMessage"/>
      <output message="toService">
        <assign to="." from="*" />
        <assign to="wsmq_qname">some_queue_name</assign>
<!-- If neither two following assign's are specified, the PrimaryDocument will be
sent
OR to perform batch sending, use the following assign:
      <assign to="wsmq_batchFilter">*</assign>
OR to send some other single document, use the following assign:
      <assign to="wsmq_document" from="SomeDocumentName/@*" />
-->
    </output>
    <input message="fromService">
      <assign to="." from="*" />
    </input>
  </operation>

  <operation name="Commit the PUT">
    <participant name="WSMQ_Commit"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
    </output>
    <input message="fromService"/>
  </operation>

  <operation name="Close Queue">
    <participant name="WSMQ_CloseQueue"/>
    <output message="toService">
      <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
      <assign to="wsmq_qname">some_queue_name</assign>
    </output>
    <input message="fromService"/>
  </operation>

  <sequence name="CloseSession">
    <operation name="Close Session">
      <participant name="WSMQ_CloseSession"/>
      <output message="toService">
        <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
      </output>
      <input message="fromService"/>
    </operation>
  </sequence>

  <onFault>
    <operation name="Make sure session is closed">
      <participant name="WSMQ_CloseSession"/>
      <output message="toService">
        <assign to="wsmq_sessionid" from="string(wsmq_sessionid)" />
      </output>
      <input message="fromService"/>
    </operation>
  </onFault>
</sequence> <!-- end of SessionOpened sequence -->
</sequence> <!-- end of OpenSession sequence -->

```

</process>

WebSphereMQ Suite Put Message Service

The WebSphereMQ Suite Put Message service is used as part of the WebSphereMQ Suite to open a queue. The following table provides an overview of the WebSphereMQ Suite Put Message service:

System Name	WSMQPutMessage
Graphical Process Modeler (GPM) categories)	All Services, Messaging
Description	Used as part of the WebSphereMQ Suite to put messages on a queue.
Business usage	Used in conjunction with the other WebSphereMQ Suite services to send or receive messages from a WebSphereMQ server.
Usage example	Used to “put” messages on a queue.
Preconfigured?	Yes – a default instance is created during install, which can be used by all business processes. Default instance name is WSMQ_PutMessage.
Requires third party files?	com.ibm.mq.jar version 5.2.0 or higher and associated message catalog property files (i.e. mqji_en_US.property). These files need to be installed using install3rdParty.
Platform availability	All supported Application platforms
Related services	WSMQOpenSession, WSMQCloseSession, WSMQOpenQueue, WSMQCloseQueue, WSMQGetMessage, WSMQCommit, WSMQBackout
Application requirements	You must have a WebSphereMQ server.
Initiates business processes?	No. This service does not invoke business processes.
Invocation	Invoke the default instance in a business process and specify the appropriate parameters.
Business process context considerations	To send a single workflow document, assign a document to the wsmq_document parameter, otherwise it defaults to use primary document. Sending multiple documents from process data can be accomplished by specifying wsmq_batchFilter.
Returned status values	<ul style="list-style-type: none">◆ Success – PutMessage service completed successfully.◆ Error – PutMessage service did not complete successfully.
Restrictions	None
Persistence level	Default
Testing considerations	While testing this service, it is recommended to turn on debugging (wsmq_debug=Yes), which provides useful information if problems occur.

Implementing the WebSphereMQ Suite Put Message Service

There are no steps required for implementing this service. The default configuration, WSMQ_PutMessage, is ready for use as soon as Application is installed. However, you can edit the service's parameter settings for specific business processes in the GPM.

Setting Up the Service in the GPM

Use the field definitions in the following table to set up the service configuration in the GPM:

Field	Description
Config	Select the name of the service configuration from the list.
Batch mode document filter (wsmq_batchFilter)	If specified, indicates batch mode sending of workflow documents. Enter an asterisk (*) for all documents including Primary Document. Any value is valid. Optional.
Buffersize override (wsmq_buffersize)	If specified, overrides the default buffersize used when streaming data. Valid values are 0-999999999. Optional.
Document tracking (wsmq_docTracking)	Specifies whether to perform document tracking. Valid values are Yes and No. Default is No. Optional.
Document to PUT (wsmq_document)	Specifies a single workflow document. Valid value is a valid workflow document. Optional.
Error Status For MQException (wsmq_errorOnMQException)	Normally when an MQException occurs, the workflow status is set to error but this allows it to be set to success so that the workflow can perform its own error process or retry logic. Valid values are Yes and No. Default is Yes. Optional.
Convert wsmq_MQMD_accountingToken from hex to byte array (wsmq_hexToByte_actToken)	Select Yes to invoke the conversion from hex to byte array. Valid values are Yes and No. Default is No. Optional.
Convert wsmq_MQMD_correlId from hex to byte array (wsmq_hexToByte_corId)	Select Yes to invoke the conversion from hex to byte array. Valid values are Yes and No. Default is No. Optional.
Convert wsmq_MQMD_groupId from hex to byte array (wsmq_hexToByte_grpId)	Select Yes to invoke the conversion from hex to byte array. Valid values are Yes and No. Default is No. Optional.
Convert wsmq_MQMD_msgId from hex to byte array (wsmq_hexToByte_msgId)	Select Yes to invoke the conversion from hex to byte array. Valid values are Yes and No. Default is No. Optional.
Maximum message size (wsmq_maxMsgSize)	Used to limit the message size. Documents larger than the maxMsgSize will be segmented. Valid values are 0-999999999. Default is zero (0). Optional.
Accounting Token (wsmq_MQMD_accountingToken)	Used to set the accounting token field of the message descriptor (MQMD). Any value is valid. Optional.

Field	Description
Application Identity Data (wsmq_MQMD_applicationIdData)	Used to set the application identity data field of the message descriptor (MQMD). Any value is valid. Optional.
Application Origin Data (wsmq_MQMD_applicationOriginData)	Used to set the application origin data field of the message descriptor (MQMD). Any value is valid. Optional.
Character Set (wsmq_MQMD_charset)	Used to set the character set field of the message descriptor (MQMD). Valid value is a valid character set identifier as defined by the MQ documentation. Default is MQCCSI_Q_MGR, character set 819 (iso-8859-1/latin1/ibm819). Optional.
Correlation Identifier (wsmq_MQMD_corId)	Used to set the correlation identifier field of the message descriptor (MQMD). Any value is valid. Default is MQCI_NONE. Optional.
Expiration Time Out (1/10 sec) (wsmq_MQMD_expiry)	Used to set the expiry field of the message descriptor (MQMD). Valid values are 0-999999999. If zero is specified, the default, MQEI_UNLIMITED, is used. Optional.
Report feedback or reason code (wsmq_MQMD_feedback)	Used to set the feedback field of the message descriptor (MQMD). The feedback field is only set if wsmq_MQMD_msgType is REPORT. Valid value is feedback or reason code value (see your MQ documentation for more information about codes). Default is MQFB_NONE. Optional.
Format (wsmq_MQMD_format)	Used to set the format field of the message descriptor (MQMD). Valid value is any valid format type (see your MQ documentation for more information about format types). Default is MQFMT_NONE. Optional.
Group Identifier (wsmq_MQMD_grpld)	Used to set the group identifier field of the message descriptor (MQMD). Any value is valid. Default is MQGI_NONE. Optional.
Message Identifier (wsmq_MQMD_msgId)	Used to set the message identifier field of the message descriptor (MQMD). Any value is valid. Default is MQMI_NONE. Optional.
Message Sequence Number (wsmq_MQMD_msgSeqNumber)	Used to set the message sequence number field of the message descriptor (MQMD). Valid values are 1-999999999. Optional.
Message Type (wsmq_MQMD_msgType)	Used to set the message type field of the message descriptor (MQMD). Optional. Valid values are: <ul style="list-style-type: none"> ◆ DATAGRAM (default) ◆ REQUEST ◆ REPORT ◆ REPLY ◆ APPL_FIRST ◆ APPL_LAST
Message Offset (wsmq_MQMD_offset)	Used to set the message offset field of the message descriptor (MQMD). Valid values are 0-999999999. Optional.

Field	Description
Message Persistence (wsmq_MQMD_persistence)	Used to set the message persistence field of the message descriptor (MQMD). Optional. Valid values are: <ul style="list-style-type: none"> ◆ Yes ◆ No ◆ Q_DEF (default)
Priority (wsmq_MQMD_priority)	Used to set the priority field of the message descriptor (MQMD). Optional. Valid values are 0-9. Default is MQPRI_PRIORITY_AS_Q_DEF.
Put Application Name (wsmq_MQMD_putApplName)	Used to set the put application name field of the message descriptor (MQMD). Any value is valid. Optional.
Put Application Type (wsmq_MQMD_putApplType)	Used to set the put application type field of the message descriptor (MQMD). Valid value is a valid PutApplicationType value. Optional.
Reply To Queue (wsmq_MQMD_replyToQ)	Used to set the reply to queue field of the message descriptor (MQMD). Any value is valid. Optional.
Reply To Queue Manager (wsmq_MQMD_replyToQM)	Used to set the reply to queue manager field of the message descriptor (MQMD). Any value is valid. Optional.
MQMF_LAST_MSG_IN_GROUP (wsmq_MQMF_lastMsgInGroup)	Used to add the MQMF_LAST_MSG_IN_GROUP Message Flag option of the message descriptor (MQMD). Valid values are Yes and No. Default is No. Optional
MQMF_MSG_IN_GROUP (wsmq_MQMF_msgInGroup)	Used to add the MQMF_MSG_IN_GROUP Message Flag option of the message descriptor (MQMD). Valid values are Yes and No. Default is No. Optional.
MQMF_SEGMENTATION_ALLOWED (wsmq_MQMF_segmentAllowed)	Used to add the MQMF_SEGMENTATION_ALLOWED Message Flag option of the message descriptor (MQMD). Valid values are Yes and No. Default is MQMF_SEGMENTATION_INHIBITED. Optional.
MQPMO_FAIL_IF QUIESCING (wsmq_MQPMO_failIfquiescing)	Used to add the MQPMO_FAIL_IF QUIESCING Put Message Option. Valid values are Yes and No. Default is No. Optional.
MQPMO_LOGICAL_ORDER (wsmq_MQPMO_logicalorder)	Used to add the MQPMO_LOGICAL_ORDER Put Message Option. Valid values are Yes and No. Default is Yes. Optional.
MQPMO_NEW_CORREL_ID (wsmq_MQPMO_newCorId)	If yes, causes the queue manager to generate a new correlation identifier in the CorrelId field of the message descriptor (MQMD). Valid values are Yes and No. Default is No. Optional.
MQPMO_NEW_MSG_ID (wsmq_MQPMO_newMsgId)	If yes, causes the queue manager to generate a new message identifier in the MsgId field of the message descriptor (MQMD). Valid values are Yes and No. Default is No. Optional.
MQPMO_SET_ALL_CONTEXT (wsmq_MQPMO_setallcontext)	Used to add the MQPMO_SET_ALL_CONTEXT Put Message Option which indicates that the user will supply all the identity and origin fields. Valid values are Yes and No. Default is No. Optional.

Field	Description
MQPMO_SET_IDENTITY_CONTEXT (wsmq_MQPMO_setidentitycontext)	Used to add the MQPMO_SET_IDENTITY_CONTEXT Put Message Option which indicates that the user will supply all the identity fields. Valid values are Yes and No. Default is No. Optional.
MQPMO_SYNCPOINT (wsmq_MQPMO_syncpoint)	If yes, adds MQPMO_SYNCPOINT Put Message Option. If no, adds MQPMO_NO_SYNCPOINT Put Message Option. Valid values are Yes and No. Default is Yes. Optional.
MQRO_COA (wsmq_MQRO_coa)	Sets the confirmation on arrival Report Option. Optional. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) (default) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA)
MQRO_COD (wsmq_MQRO_cod)	Sets the confirmation on delivery Report Option. Optional. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) (default) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA)
MQRO_DISCARD_MSG (wsmq_MQRO_discard)	If yes, adds the MQRO_DISCARD_MSG Report Option. Valid values are Yes and No. No assumes MQRO_DEAD_LETTER_Q. Optional.
MQRO_EXCEPTION (wsmq_MQRO_exception)	Sets the exception Report Option. Optional. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) (default) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA)
MQRO_EXPIRATION (wsmq_MQRO_expiration)	Sets the expiration Report Option. Optional. Valid values are: <ul style="list-style-type: none"> ◆ No Report (NOT_USED) (default) ◆ Report without data (NO_DATA) ◆ Report with partial data (WITH_DATA) ◆ Report with full data (FULL_DATA)
MQRO_NAN (wsmq_MQRO_nan)	If yes, adds the MQRO_NAN Report Option. Valid values are Yes and No. Default is No. Optional.
MQRO_PAN (wsmq_MQRO_pan)	If yes, adds the MQRO_PAN Report Option. Valid values are Yes and No. Default is No. Optional.

Field	Description
MQRO_PASS_CORREL_ID (wsmq_MQRO_passCorId)	If yes, adds the MQRO_PASS_CORREL_ID Report Option. Valid values are Yes and No. Default is No, which sets MQRO_COPY_MSG_ID_TO_CORREL_ID. Optional.
MQRO_PASS_MSG_ID (wsmq_MQRO_passMsgId)	If yes, adds the MQRO_PASS_MSG_ID Report Option. Valid values are Yes and No. Default is No, which sets MQRO_NEW_MSG_ID. Optional.
No Segment Splitting (wsmq_noSegSplit)	If messages need to be split (see wsmq_maxMsgSize), they will be sent as separate messages instead of segments. Valid values are Yes and No. Default is No. Optional.
Queue Name (wsmq_qname)	The name of a previously opened queue used to PUT messages. Any value is valid. Required.

Parameters Passed From Business Process to Service

The following table contains the parameters passed from the business process to the WebSphereMQ Suite Put Message service:

Parameter	Description
All GPM parameters	See <i>Setting Up the Service in the GPM</i> .
wsmq_sessionid (wsmq_sessionid)	The session identifier to use from a previously established session. Valid value is the value from a previously opened session. Required.

Parameters Passed From Service to Business Process

The following table contains the parameters passed from the WebSphereMQ Suite Put Message service to the business process. The values of the parameters are added to the original document tag where necessary, as shown in this example that uses the sentMsgId parameter:

```
<ProcessData> <!-- before PUT -->
```

```
<ProcessData> <!-- after PUT -->
```

```
    sentMsgId
```

```
    sentMsgId
```

Parameter	Description
sentMsgId	The message identifier that was sent with the message. This value was either assigned by the user or assigned by the queue manager. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.

Parameter	Description
sentCorId	The correlation identifier that was sent with the message. This value was either assigned by the user or assigned by the queue manager. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
sentGrpId	The group identifier that was sent with the message. This value was either assigned by the user or assigned by the queue manager. Because this value could contain embedded nulls, it is a hex representation of the string so that no characters are lost.
nextMsgSeqNum	If sending multiple messages as part of a logical group but MQPMO_LOGICAL_ORDER is not used and MQMF_MSG_IN_GROUP and/or MQMF_LAST_MSG_IN_GROUP is specified, the adapter will automatically take care of incrementing the message sequence number from its initial value but only if segmenting is required. In this scenario, the value of this variable should be used in the wsmq_MQMD_msgSeqNumber parameter to set the next initial message sequence number.
nextMsgOffset	If sending multiple messages as part of a logical group but MQPMO_LOGICAL_ORDER is not used, the adapter will automatically take care of incrementing the message offset number from its initial value but only if segmenting is required. In this scenario, the value of this variable should be used in the wsmq_MQMD_offset parameter to set the next initial message offset value.
WSMQ/CompletionCode	Only set if an MQException occurs and indicates the completion code.
WSMQ/ReasonCode	Only set if an MQException occurs and indicates the reason code.

Business Process Example

The following example illustrates how the WebSphereMQ Suite Put Message service can be used in a business process:

WSDL Service

The following table provides an overview of the WSDL service:

System name	WSDL Service
Graphical Process Modeler (GPM) categories	All Services, Internet B2B > SOAP
Description	Web Services Description Language (WSDL) is a standard used to provide technical information about a Web service, including its description, how to connect to it, and what the service requires and returns. The WSDL Reader Service leverages this information to automate the connection to the Web service.
Business usage	The business value of this service is to make use of Web services easier for a business user.
Usage example	You discover a Web service to use (for example, by browsing UDDI directories) and download the WSDL describing it. The WSDL XML document should be checked in to and versioned by the application as a WSDL document object. The business use is at runtime: when the business process goes to use the Web service, it queries the checked-in WSDL document for connection information.
Preconfigured?	Yes
Requires third party files?	None
Platform availability	All supported application platforms
Related services	This service is designed to be used in conjunction with the business process SendSOAPRequest. The WSDL service is designed to provide the SendSOAPRequest business process with SOAP connection information.
Application requirements	A valid WSDL document properly installed in the application. The specified operation exists in the specified WSDL document.
Initiates business processes?	No
Invocation	Runs as part of a business process
Business process context considerations	No
Returned status values	<ul style="list-style-type: none">◆ Success – The WSDL object is found and the operation is located within the definition.◆ Error – Other result.
Restrictions	No
Persistence level	Full
Testing considerations	None

How the WSDL Service Works

The WSDL service queries WSDL documents that have been checked in to the application for connection information during a business process. Before you can use the WSDL service, you need to check in the WSDL documents that you will use in your business process. WSDL documents can be checked in or created manually using the **Deployment > Web Services > WSDL Check In** menu option in the application.

Implementing the WSDL Service

To implement the WSDL service, complete the following process:

1. Create a WSDL service configuration.
2. Configure the WSDL service.
3. Use the WSDL service in a business process.

Configuring the WSDL Service

The following table describes the fields used to configure the WSDL service in the GPM:

Field	Description
Config	Name of the service configuration.
Operation	Operation to process. Required.
WSDLFile	WSDL filename to read. Optional. Note: Either WSDLFile or wsdlName must be entered. If both fields are filled in, the wsdlName entry is used and the wsdlFile entry is ignored.
wsdlName	Name given to the WSDL document when it is checked in to the application. Note: Either WSDLFile or wsdlName must be entered. If both fields are filled in, the wsdlName is used and the wsdlFile is ignored.
wsdlVersion	Version of the WSDL document to use, if more than one exists. Optional.

Output from Service to Business Process

The following table describes the output from the WSDL service to the business process:

Parameter Name (BPML Element Value)	Description
End Point URL (SOAPRequestURL)	The URL to configure the SOAP outbound client with. Required.
SOAP Action (SOAP_ACTION)	The header value SOAP-ACTION. Optional.

Output from Business Process to Service

The following table describes the output from the business process to the WSDL service:

Parameter Name (BPML Element Value)	Description
WSDL Operation name (operation)	Operation to process. Required.
WSDL Filename (wsdlFile)	WSDL filename to read. Optional. Note: Either wsdlFile or wsdlName must be entered. If both fields are filled in, the wsdlName entry is used and the wsdlFile entry is ignored.
WSDL Object Name (wsdlName)	Name given to the WSDL document when it is checked in to the application. Note: Either wsdlFile or wsdlName must be entered. If both fields are filled in, the wsdlName is used and the wsdlFile is ignored.
WSDL Version (wsdlVersion)	Version of the WSDL document to use, if more than one exists. Optional.

XML Digital Signature Service (Build 4315 - Build 4318)

The following table provides an overview of the XML Digital Signature service:

Service Name	XML Digital Signature Service
System name	XMLDSigService
Graphical Process Modeler (GPM) Category	All Services
Description	Use the XML Digital Signature service to compose and verify digital signatures.
Business usage	Use this service to create enveloped, enveloping, detached, and a combination of all three signatures.
Usage example	A business process that needs a document to be digitally signed or verified can invoke this service by passing the required parameters.
Preconfigured?	No
Requires third party files?	Yes. Requires xss4j.jar. This is preloaded in the Application.
Platform availability	All supported Application platforms
Related services	N/A
Application requirements	N/A
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	You must be familiar with the Internal Service (WF/BP parameters) that invokes this adapter. The WF parameters are the values passed into the Internal Service and BP parameters are the values specified within the business process code.
Returned status values	<ul style="list-style-type: none">◆ signRequest◆ verifyRequest
Restrictions	None
Persistence level	System Default
Testing considerations	You should use the correct certificates for signing. The most common problem encountered is that certificates used for signing are not created with a storepass value and a keypass value of integrator. If you receive an error with this condition, see your system administrator.
Restrictions	None

How the XML Digital Signature Service Works

The XML Digital Signature service signs or verifies the XML signature. It provides integrity and confidentiality of XML documents and messages.

The XML Digital Signature service in the Application supports the following types of XML signatures:

Enveloped (default) - signature of either an entire document or a document fragment where the XML signature is embedded within the signed document.

Enveloping - signature where signed data is embedded within XML signature structure.

Detached - signature where the signed entities are not attached to the actual signature fragment.

Note: In the Application, a detached signature type signs on the detached workflow document. The Reference URI of the detached document is the document ID.

Combination (combination of enveloped, enveloping, and detached)

Implementing the XML Digital Signature Service

To implement the XML Digital Signature service, complete the following tasks:

1. Create an XML Digital Signature service configuration. For information, see *Managing Services and Adapters*.
2. Configure the XML Digital Signature service. For information, see *Configuring the XML Digital Signature Service* on page 1676.
3. Use the XML Digital Signature service in a business process.

System Administrator Tasks

The following procedure describes the system administrator tasks for XML Digital Signature service.

Importing a KeyCert into Application

1. Login to Application.
2. Select **Trading Partner** -> **Digital Certificates** -> **Trusted**.
3. Select **New Certificate** under Check in.
4. Select the certificate and click **Next**.
5. Enter the Certificate Name and click **Next**.
6. Review and click **Finish**.
7. You can use this certificate in your BPML associated with the appropriate field (signCertificateIdentifier).

Configuring the XML Digital Signature Service

To configure the XML Digital Signature service, you must specify settings for the following fields:

Field	Description
Name	XML Digital Signature Service
Description	Signs and validates XML digital signatures
Select a Group	Select one of the following options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .

Output from Business Process to Service

The following table contains the parameters passed from the business process to the XML Digital Signature service when it invokes with the output message set to `signRequest`:

Parameter	Description
action	Required. The required action. The value can be a valid string. Valid value - sign.
signatureType	Required. The type of signature. Valid values are: <ul style="list-style-type: none">◆ enveloped (default)◆ enveloping◆ detached◆ combination of enveloped, enveloping, and detached
signCertificateIdentifier	Required. The alias of a private key in the certificate.
certificateIdentifier	Optional. The alias of a public key in the certificate. When this parameter is used during signing, the <code>KeyInfo</code> element must be included in the signature.
nodeToSign	Optional. Indicates the node that needs to be signed. This parameter is used when signing XML document and the node exists in the document to be signed. If this parameter is not specified, the whole document is signed. Multiple nodes can be specified using comma (,) as delimiter. When signing with enveloped signature, the node to be signed should contain ID attribute. Valid node names - node1, node2

Parameter	Description
Transforms	<p>Optional. The required Transforms to be used when signing. If omitted when signing with enveloped signature type, the enveloped-signature Transform will be used.</p> <p>If omitted when signing non-XML document with enveloping signature type, the base64 Transform will be used.</p> <p>An error will be thrown when the transform algorithm is invalid in xss4j.</p> <p>A valid example using Transform:</p> <pre data-bbox="472 470 935 617"><Transforms> <Transform Algorithm="..."> </Transform> </Transforms></pre>
documents	<p>Optional. Used to sign multiple documents or when primary document is empty. This node contains a list of document nodes.</p> <p>This parameter is required when primary document is empty.</p> <p>Each document node contains:</p> <ul data-bbox="472 785 1377 1010" style="list-style-type: none"> ◆ documentID - Required for all signature types. ◆ nodeToSign - Required when signing XML document and optional for all signature types. ◆ signatureType - Required for combination signature. ◆ Transforms/Transform - Optional for all signature types and can contain multiple Transform nodes with different algorithms. <p>A valid example:</p> <pre data-bbox="472 1062 841 1755"><documents> <document> <documentID> xxx </documentID> <signatureType> xxx </signatureType> <nodeToSign> node1, node2 </nodeToSign> <Transforms> <Transform Algorithm="..."> </Transform> </Transforms> </document> </documents></pre>

The following table contains the parameters passed from the business process to the XML Digital Signature service when it invokes with the output message set to `verifyRequest`:

Parameter	Description
action	Required. The required action. The value can be a valid string. Valid value - verify.
certificateIdentifier	Optional. The alias of a public key in the certificate. If certificateIdentifier is not present, the certificate information is retrieved from the KeyInfo element of signature.
documents	Optional. Used for detached signature verification. It contains a list of document nodes and each document contains one documentID. The sequence of the detached document list should follow the reference sequence in XML signature. This parameter is not applicable for verifying enveloped and enveloping signature. A valid example: <pre><documents> <document> <documentID> xxx </documentID> </document> </documents></pre>

Business Process Examples

The following example business processes illustrate using the XML Digital Signature service:

Example Business Process 1

The following BPML signs the document based on the parameters passed from BPML to the XML Digital Signature service.

```
<process name="xmldsig_enveloped">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="signatureType">enveloped</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
```

```

    <participant name="XMLDSigService"/>
    <output message="verifyRequest">
      <assign to="." from="*" />
      <assign to="action">verify</assign>
      <assign to="certificateIdentifier">test_rsa_pub</assign>
    </output>
    <input message="verifyResponse">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

Example Business Process 2

The following BPML shows how to sign the Primary Document and add the KeyInfo element within the Signature element by including the certificateIdentifier parameter in the signing request. The example also includes how to construct the Transforms node.

```

<process name="xmldsig_enveloped_transform_keyinfo">
  <sequence>
    <assign
to="temp/@Algorithm">http://www.w3.org/2000/09/xmldsig#enveloped-signature</assign>
    <assign to="Transforms/Transform" from="temp/@*" />

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="signatureType">enveloped</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
        <assign to="Transforms" from="Transforms/node()" />
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
        <assign to="." from="*" />
        <assign to="action">verify</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>

```

Example Business Process 3

The following BPML shows how to sign particular nodes in the Primary Document. The nodes to be signed are delimited by comma (,). The signature type is enveloped.

```
<process name="xmldsig_enveloped_nodetosign">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="nodeToSign">value1,value2</assign>
        <assign to="signatureType">enveloped</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
        <assign to="." from="*" />
        <assign to="action">verify</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

Example Business Process 4

The following BPML shows how to sign particular nodes in the Primary Document. The nodes to be signed are delimited by comma (,). The signature type is enveloping.

```
<process name="xmldsig_enveloping_nodetosign">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="nodeToSign">value1,value2</assign>
        <assign to="signatureType">enveloping</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```



```

    </input>
  </operation>

  <operation name="VerifyMessage">
    <participant name="XMLDSigService"/>
    <output message="verifyRequest">
      <assign to="." from="*" />
      <assign to="action">verify</assign>
      <assign to="certificateIdentifier">test_rsa_pub</assign>
    </output>
    <input message="verifyResponse">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

Example Business Process 5

This following input file and BPML shows how to sign multiple documents passed from “documents/document” parameter. The nodes to be signed are delimited by comma (,). The signature type is enveloping.

Input file:

```

<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <nodeToSign>value1,value2</nodeToSign>
    <Transforms>
      <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
        <XPath>descendant-or-self::Contract</XPath>
      </Transform>
    </Transforms>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
  </document>
</documents>

```

BPML:

```

<process name="xmldsig_enveloping_documents">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">documents</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

```

```

<operation>
  <participant name="ReleaseService"/>
  <output message="releaseRequest">
    <assign to="TARGET">PrimaryDocument</assign>
  </output>
  <input message="releaseResponse">
  </input>
</operation>

<operation name="SignMessage">
  <participant name="XMLDSigService"/>
  <output message="signRequest">
    <assign to="." from="*"></assign>
    <assign to="action">sign</assign>
    <assign to="signatureType">enveloping</assign>
    <assign to="signCertificateIdentifier">test_rsa_priv</assign>
    <assign to="documents" from="documents/node()"></assign>
  </output>
  <input message="signResponse">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation>
  <participant name="ReleaseService"/>
  <output message="releaseRequest">
    <assign to="TARGET" from="'documents'"></assign>
  </output>
  <input message="releaseResponse">
  </input>
</operation>

<operation name="VerifyMessage">
  <participant name="XMLDSigService"/>
  <output message="verifyRequest">
    <assign to="." from="*"></assign>
    <assign to="action">verify</assign>
    <assign to="certificateIdentifier">test_rsa_pub</assign>
  </output>
  <input message="verifyResponse">
    <assign to="." from="*"></assign>
  </input>
</operation>

</sequence>
</process>

```

Example Business Process 6

The following input file and BPML shows how to sign multiple documents passed from “documents/document” parameter, Transforms/Transform/XPath is used to sign specific node. The signature type is detached.

Input file:

```

<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <Transforms>
      <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
        <XPath>descendant-or-self::Contract</XPath>
      </Transform>
    </Transforms>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
  </document>
</documents>

```

BPML:

```

<process name="xmldsig_enveloping_documents">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">documents</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET">PrimaryDocument</assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*"></assign>
        <assign to="action">sign</assign>
        <assign to="signatureType">detached</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="documents" from="documents/node()"></assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">

```

```

    <assign to="TARGET" from="'documents'"></assign>
  </output>
  <input message="releaseResponse">
  </input>
</operation>

<assign to="document/documentID">sgconrad:-628e3b67:11569be511e:-6d7a</assign>
<assign to="documents/document" from="document/node()" append="true"></assign>
<assign to="document/documentID">sgconrad:-628e3b67:11569be511e:-682f</assign>
<assign to="documents/document" from="document/node()" append="true"></assign>

<operation name="VerifyMessage">
  <participant name="XMLDSigService"/>
  <output message="verifyRequest">
    <assign to="." from="*"></assign>
    <assign to="action">verify</assign>
    <assign to="certificateIdentifier">test_rsa_pub</assign>
  </output>
  <input message="verifyResponse">
    <assign to="." from="*"></assign>
  </input>
</operation>

</sequence>
</process>

```

Example Business Process 7

The following input file and BPML shows how to sign multiple documents passed from “documents/document” parameter. The signature type is combination.

Input file:

```

<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <signatureType>enveloped</signatureType>
    <nodeToSign>value1,value2</nodeToSign>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
    <signatureType>enveloping</signatureType>
    <Transforms>
      <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64">
      </Transform>
    </Transforms>
  </document>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75bc</documentID>
    <signatureType>detached</signatureType>
    <nodeToSign>node1,node2</nodeToSign>
  </document>
</documents>

```

BPML:

```

<process name="xmldsig_enveloping_documents">

```

```

<sequence>
  <operation name="XML Encoder">
    <participant name="XMLEncoder"/>
    <output message="XMLEncoderTypeInputMessage">
      <assign to="output_to_process_data">YES</assign>
      <assign to="mode">xml_to_process_data</assign>
      <assign to="root_element">documents</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation>
    <participant name="ReleaseService"/>
    <output message="releaseRequest">
      <assign to="TARGET">PrimaryDocument</assign>
    </output>
    <input message="releaseResponse">
    </input>
  </operation>

  <operation name="SignMessage">
    <participant name="XMLDSigService"/>
    <output message="signRequest">
      <assign to="." from="*"></assign>
      <assign to="action">sign</assign>
      <assign to="signatureType">combination</assign>
      <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      <assign to="documents" from="documents/node()"></assign>
    </output>
    <input message="signResponse">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation>
    <participant name="ReleaseService"/>
    <output message="releaseRequest">
      <assign to="TARGET" from="'documents'"></assign>
    </output>
    <input message="releaseResponse">
    </input>
  </operation>

  <assign to="document/documentID">sgconrad:-628e3b67:11569be511e:-6d7a</assign>
  <assign to="documents/document" from="document/node()" append="true"></assign>

  <operation name="VerifyMessage">
    <participant name="XMLDSigService"/>
    <output message="verifyRequest">
      <assign to="." from="*"></assign>
      <assign to="action">verify</assign>
      <assign to="certificateIdentifier">test_rsa_pub</assign>
    </output>

```

```

        <input message="verifyResponse">
            <assign to="." from="*"></assign>
        </input>
    </operation>

</sequence>
</process>

```

XML Digital Signature Service Examples

The following example signature types illustrate using the XML Digital Signature service:

Example of Enveloped Signature

The following example shows the enveloped signature type:

```

<?xml version="1.0" encoding="UTF-8"?>
<test>
    <value1 ID="1">
testval1</value1>
    <value2 ID="2">
testval2</value2>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
    <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="">
    <Transforms>
    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
fgWg+5VNk9ZqUy86IOIE3QS+bGQ=</DigestValue>
    </Reference>
    </SignedInfo>
    <SignatureValue>
MYWSLB0T9qX+Qu2BnMwPgc4KmKT5pws0M9GtQQsMGYT5rxXCMe2qeMKAV65WoY0ib7mTxjj9
nChQsdyN1yYYn6nRvvKl9lX8LWVNpJj58lE7/alohe8/+0hOAWKxBsiFDhBbUC0jHG1+Qp2f
M3KAe2kyBUPjN57isAmmH64y/LI= </SignatureValue>
    <KeyInfo>
    <KeyValue>
    <RSAKeyValue>
    <Modulus>
kFhwg4m9hjFmrlxVR3w0XmYx7fgGsoh+aelmXlzCug5gRV0t0XeSpaeoX1jXu6gacJ
Vl/p0lNs+av+iviDKmS94LDPJtjAcl7C9dZbbt39N+/2S9WBAtJGxk5MOIu0aab50D
UfK55mUbpsZzwoVQrisW+KArnWlbrUP5xWXsnwM= </Modulus>
    <Exponent>
AQAB</Exponent>
    </RSAKeyValue>
    </KeyValue>
    <X509Data>
    <X509IssuerSerial>
    <X509IssuerName>

```

```

CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509IssuerName
>
  <X509SerialNumber>
1190704157</X509SerialNumber>
  </X509IssuerSerial>
  <X509SubjectName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509SubjectName
e>
  <X509Certificate>
MIICBjCCAW8CBEB4tB0wDQYJKoZIhvcNAQEFBQAwSTELMCMGCSqGSIB3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvbTELMkA1UEBhMCU0cxEzARBgNVBAMTCnNlcmVhYV9yc2EwIBcNMDCwOTI1
MDCwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEkxJTAjBggqhkkiG9w0BCQEWFnNlcmVhYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTA1NHMRMwEQYDQVQDEWpZzXJlcmVhYV9saUBzdGVyY29tbS5jb20
AQUAA4GNADCBiQKBGQCCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5Klp6hf
WNe7qBpwlWX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL111tu3f037/ZL1YEC0kZeTkW4i7RppvnQNQ
UrnMZRumxnPCChVCuKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIB3DQEBBQUAA4GBAC+7glCs
TKBSURkwmBA4k/SYVOOhhz3VkBX0he3r1/Vd6Qk8I1RjWQj5AT8e40gz+vg00GvjaYAx70bvIGqn
yYE/VVJJOG5Zw6Tott69Dx4AOCrmbZb96z0AjlCEI3017U1h+9+Uo2h5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
  </X509Data>
  </KeyInfo>
  </Signature>
</test>

```

Example of Enveloped Signature with particular nodes

The following example shows the enveloped signature type with particular nodes:

```

<?xml version="1.0" encoding="UTF-8"?>
<test>
  <value1 ID="1">
testval1</value1>
  <value2 ID="2">
testval2</value2>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
  <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <Reference URI="#1">
  <Transforms>
  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>
XEZm5p6sjqI+Gt61BHQTOwTl2Nw=</DigestValue>
  </Reference>
  <Reference URI="#2">
  <Transforms>
  <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>
1k1M9eAUMUWM4FP40A1/YC9qS/Y=</DigestValue>
  </Reference>
  </SignedInfo>

```

```

    <SignatureValue>
UmZlQ5g4ummBZu4IRg+BHDpZeeX4ix1GXODhyG+f5tvw+4ZDKIGn8qm5wdAj3+/wJS84Lx/V
hRgjyaucMjM3qEoMAEXEWYRTag0PdeOmfrSUffictYcTaGpPUYjDiWfKyFXAlCL5XmSeBW5s
MGgy5Bi/Y0HB1gw1mu+Jsh6FsGM= </SignatureValue>
    <KeyInfo>
    <KeyValue>
    <RSAKeyValue>
    <Modulus>
kFhwg4m9hjFmrlxVR3w0XmYx7fgGsoh+aelmXlzCug5gRV0t0XeSpaeoX1jXu6gacJ
Vl/p0lNs+av+iviDKmS94LDPJtjAc17C9dZbbt39N+/2S9WBAtJGXk5M0Iu0aab50D
UFK55mUbpsZzwoVQrisW+KArnWlbrUP5xWXsnwM= </Modulus>
    <Exponent>
AQAB</Exponent>
    </RSAKeyValue>
    </KeyValue>
    <X509Data>
    <X509IssuerSerial>
    <X509IssuerName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509IssuerName
>
    <X509SerialNumber>
1190704157</X509SerialNumber>
    </X509IssuerSerial>
    <X509SubjectName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509SubjectNam
e>
    <X509Certificate>
MIICBjCCAW8CBEb4tB0wDQYJKoZIhvcNAQEFBQAwSTElMCMGCSqGSIB3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvbTELMakGA1UEBhMCU0cxZARBGNVBAWTCnNlcmVlYV9yc2EwIBcnMDcwOTI1
MDCwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEkxJTAjBjkqhkig9w0BCQEFwFnNlcmVlYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTA1NHMRMwEQYDQDEwpzZXJlbmFfc3R5bGUfMA0GCSqGSIB3DQEB
AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5Klp6hf
WNe7qBpwlWX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL111tu3f037/ZL1YEC0kZeTkw4i7RppvnQNQ
UrnMZRumxnPChVCuKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIB3DQEBBQUAA4GBAC+7glCs
TKBSURkwmBA4k/SYVOOhhz3VkBX0he3r1/Vd6Qk8I1RjWQj5AT8e40gz+vg00GvjaYAx70bVIGqn
yYE/VVJJOJ5Zw6Tott69Dx4AOCrmbZb96z0AjlceI3017U1h+9+Uo2h5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
    </X509Data>
    </KeyInfo>
    </Signature>
</test>

```

Example of Enveloping Signature

The following example shows the enveloping signature type:

```

<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="#test">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
ILdA0tPr6GHE9C2/aLX4GQZc3XI=</DigestValue>

```



```

    <value1 ID="1">
testval1</value1>
    </Object>
    <Object xmlns="" Id="value2">
    <value2 ID="2">
testval2</value2>
    </Object>
</Signature>

```

Example of Enveloping Signature with non-XML Input File

The following example shows the enveloping signature type with non-XML input file:

```

<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="#sgconrad:-7cd5f978:1159315afbc:-6124">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
UWdFQL/JwoDirPg/AJdp+m5+bT4=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>
cAtHLlmNUVRmWzn4mDvWkcRvFDoKOkes+gMsnC4pHAKclg99j+e2xxR0SsE5HnvNPEH3IrwT
GZyaTXVlx3UTaXlC+215t0mW4CYn4nyZpwJTbM18pRZq8tijquydg4roZz/yawz856uow3KH
z+khz0uwt78GzwQXVyqQymyVrQk= </SignatureValue>
    <Object xmlns="" Encoding="base64" Id="sgconrad:-7cd5f978:1159315afbc:-6124">
dGhpcyBpcyB0ZXN0 </Object>
</Signature>

```

Example of Detached Signature

The following example shows the detached signature type:

```

<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="sgconrad:31e5343c:1158d3b080f:-75fc">
    <Transforms>
    <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
    <XPath>
descendant-or-self::Contract</XPath>
    </Transform>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
1ybLCHRnYSGKCoswkUOuD650Mr0=</DigestValue>
    </Reference>
    <Reference URI="sgconrad:-3c3ab664:1158cfc1d5d:-5431">
    <Transforms>
    <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">

```

```

    <XPath>
descendant-or-self::FILLER</XPath>
    </Transform>
  </Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>
2jmj715rSw0yVb/vlWAYkK/YBwk=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>
CBML9dFb/hEQXXR7oYfTuu4qit/VhUjwIfvPhSUQTQg0j+BFiTzFwNZaCJkZGswxDnSKhH1p
CuLn/Fpzl2CJpNduDU0Ff0pstd7MITS010/IvhDVS+Tf6WiYkN8UYTCkJeg063z1bW+15mR1
Z25jCs0gWO9qESthX34qXRi7ii0= </SignatureValue>
</Signature>

```

Example of Combination Signature of Enveloped/Enveloping/Detached

The following example shows the combination signature type:

```

<?xml version="1.0" encoding="UTF-8"?>
<test>
  <value1 ID="1">
testval1</value1>
  <value2 ID="2">
testval2</value2>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
    <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <Reference URI="#1">
    <Transforms>
    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
IpBpovbT2WG7C+gTME1Np/V2fgo=</DigestValue>
    </Reference>
    <Reference URI="#2">
    <Transforms>
    <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
pFXQ3ZZv4Fivm2MFs6vpfEanEDI=</DigestValue>
    </Reference>
    <Reference URI="#sgconrad:-56000361:115d676b12e:-7988">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
7bpj9pPMJpsJw1OJ1b2jsrhxYMY=</DigestValue>
    </Reference>
    <Reference URI="sgconrad:-56000361:115d676b12e:-795f">
    <Transforms>
    <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
    <XPath>
descendant-or-self::node1</XPath>

```

```

</Transform>
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>
2gFHdr03uDeDqwcxGveD+uYDIjM=</DigestValue>
</Reference>
<Reference URI="sgconrad:-56000361:115d676b12e:-795f">
<Transforms>
<Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
<XPath>
descendant-or-self::node2</XPath>
</Transform>
</Transforms>
<DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<DigestValue>
nn7t7PJs5RqDp1BKZ4j1BxhX2ik=</DigestValue>
</Reference>
</SignedInfo>
<SignatureValue>
Ui7XYcZNkXG+9OqNgKrcRJoyCuhpoRtVmFnXVOMf8aAuGXZw3FwFxz7VLKv9c1K8ZUNW9vCs
G4EpahlCS4AcpbVBwv00HvkhA11/tqYYB9kRK/wM4cb6sN5ULbQ4Ab0j9xyFKOQ6sr2MawOx
fdNEes6XAHbpWZvxKDR4vWxAFnE= </SignatureValue>
<Object xmlns="" Encoding="base64" Id="sgconrad:-56000361:115d676b12e:-7988">
dGhpcyBpcyBhbiBpbnZhbGlkIHRlc3QgZG9jIGZvciB4bWwgZHNpZyBzZXJ2aWNlLgo=</Object>
</Signature>
</test>

```

XML Digital Signature Service (Build 4319 or higher)

The following table provides an overview of the XML Digital Signature service:

Service Name	XML Digital Signature Service
System name	XMLDSigService
Graphical Process Modeler (GPM) Category	All Services
Description	Use the XML Digital Signature service to compose and verify XML digital signatures.
Business usage	Use this service to create enveloped, enveloping, detached, and a combination of all three signatures. Note: Generating a detached signature in the application means generating a detached signature on the workflow document.
Usage example	A business process that needs a document to be digitally signed or verified can invoke this service by passing the required parameters.
Preconfigured?	No
Requires third party files?	Apache XML Security 1.3.0 (xmlsec-si.jar). This is preloaded in the Application install.
Platform availability	All supported Application platforms
Related services	N/A
Application requirements	N/A
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	You must be familiar with the Internal Service (WF/BP parameters) that invokes this adapter. The WF parameters are the values passed into the Internal Service and BP parameters are the values specified within the business process code.
Returned status values	<ul style="list-style-type: none">◆ signRequest◆ verifyRequest
Restrictions	None
Persistence level	System Default
Testing considerations	You should use the correct certificates for signing. The most common problem encountered is that certificates used for signing are not created with a storepass value and a keypass value of integrator. If you receive an error with this condition, see your system administrator.

How the XML Digital Signature Service Works

The XML Digital Signature service signs or verifies the XML signature. It provides integrity and confidentiality of XML documents and messages.

The XML Digital Signature service in the Application supports the following types of XML signatures:

Enveloped (default) - signature of either an entire document or a document fragment where the XML signature is embedded within the signed document.

Enveloping - signature where signed data is embedded within XML signature structure.

Detached - signature where the signed entities are not attached to the actual signature fragment.

Note: In the Application, a detached signature type signs on the detached workflow document. The Reference URI of the detached document is the document ID.

Combination (combination of enveloped, enveloping, and detached)

Implementing the XML Digital Signature Service

To implement the XML Digital Signature service, complete the following tasks:

1. Create an XML Digital Signature service configuration. For information, see *Managing Services and Adapters*.
2. Configure the XML Digital Signature service. For information, see *Configuring the XML Digital Signature Service* on page 1696.
3. Use the XML Digital Signature service in a business process.

System Administrator Tasks

The following procedure describes the system administrator tasks for XML Digital Signature service.

Importing a KeyCert into Application

1. Login to Application.
2. Select **Trading Partner** -> **Digital Certificates** -> **Trusted**.
3. Select **New Certificate** under Check in.
4. Select the certificate and click **Next**.
5. Enter the Certificate Name and click **Next**.
6. Review and click **Finish**.
7. You can use this certificate in your BPML associated with the appropriate field (signCertificateIdentifier).

Configuring the XML Digital Signature Service

To configure the XML Digital Signature service, you must specify settings for the following fields:

Field	Description
Name	XML Digital Signature Service
Description	Signs and validates XML digital signatures
Select a Group	Select one of the following options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: For more information about groups, see <i>Managing Services and Adapters</i> .

Output from Business Process to Service

The following table contains the parameters passed from the business process to the XML Digital Signature service when it invokes with the output message set to `signRequest`:

Parameter	Description
action	Required. The required action. The value can be a valid string. Valid value - sign.
signatureType	Required. The type of signature. Valid values are: <ul style="list-style-type: none">◆ enveloped (default)◆ enveloping◆ detached◆ combination of enveloped, enveloping, and detached
signCertificateIdentifier	Required. The alias of a private key in the certificate.
certificateIdentifier	Optional. The alias of a public key in the certificate. When this parameter is used during signing, the <code>KeyInfo</code> element must be included in the signature.

Parameter	Description
nodeToSign	<p>Optional. This parameter contains the list of the XML nodes to be signed or valid xpointer path. It is only used when signing an XML document, and the node must exist in the document to be signed. If this parameter is not specified, it implies that you wish to sign the whole XML document. Multiple nodes or xpointer pathes can be specified by delimiting with a comma (,).</p> <p>When signing with enveloped signature, the node to be signed must contain attribute "ID" or "Id" or "id". When signing with detached signature, XPath transform will be automatically added to the reference.</p> <p>Note: The xpointer path only works with enveloped signature.</p> <p>Valid node names - node1, node2</p>
ds:Transforms	<p>Optional. The ds:Transforms node list to be used for transforming original data before calculating digest. The list of the transforms will be appended to the Reference node. If omitted when signing with enveloped signature, the enveloped-signature Transform will be used. If omitted when signing non-xml document with enveloping signature, the base64 Transform will be used. If omitted when signing specific node with detached signature, the XPath Transform will be used.</p> <p>An error will be thrown when the transform algorithm is not supported by Apache XML Security.</p> <p>A valid example using Transform:</p> <pre><ds:Transforms xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> < ds:Transform Algorithm='...'> </ ds:Transform> </ ds:Transforms></pre>

Parameter	Description
documents	<p>Optional. Used to sign multiple documents or when primary document is empty. This node contains a list of document nodes.</p> <p>Each document node contains:</p> <ul style="list-style-type: none"> ◆ documentID - Mandatory for all signature types. ◆ nodeToSign - Optional for all signature types and is only used when signing xml documents. ◆ signatureType - Mandatory for combination signature. ◆ ds:Transforms/ds:Transform - Optional for all signature types and can contain multiple ds:Transform nodes with different algorithms. <p>When signing with enveloped signature, only one document can be signed. When signing with combination signature, only one document can be signed with enveloped signature type.</p> <p>A valid example:</p> <pre> <documents> <document> <documentID> xxx </documentID> <signatureType> xxx </signatureType> <nodeToSign> node1, node2 </nodeToSign> <ds:Transforms xmlns:ds="http://www.w3.org/2000/09/xmldsig#"> < ds :Transform Algorithm="..."> </ ds :Transform> </ ds :Transforms> </document> </documents> </pre>
sigUnderNodeToSign	<p>Optional. Used to put the signature under the specific node to be signed. It is only valid when the value of signatureType is enveloped. This parameter has two possible string values - true and false. The default value is false.</p>
digestMethod	<p>Optional. Used to specify a different digestMethod than the default value. Possible value is a valid digest method URI. If this parameter is not set, the default digest method is http://www.w3.org/2000/09/xmldsig#sha1.</p> <p>An error will be thrown when the digest algorithm is not supported by Apache XML Security.</p>

Parameter	Description
signatureMethod	Optional. Used to specify a different signatureMethod than the default value. Possible value is a valid signature method URI. For example, when signing with DSA key, the signatureMethod will be http://www.w3.org/2000/09/xmldsig#dsa-sha1 . If this parameter is not set, the default signature method is http://www.w3.org/2000/09/xmldsig#rsa-sha1 . An error will be thrown when the signature algorithm is not supported by Apache XML Security.
canonicalizationMethod	Optional. Used to specify a different canonicalizationMethod than the default value. Possible value is a valid canonicalization method URI. If this parameter is not set, the default canonicalization method is http://www.w3.org/TR/2001/REC-xml-c14n-20010315 . An error will be thrown when the canonicalization algorithm is not supported by Apache XML Security.

The following table contains the parameters passed from the business process to the XML Digital Signature service when it invokes with the output message set to `verifyRequest`:

Parameter	Description
action	Required. The required action. The value can be a valid string. Valid value - verify.
certificateIdentifier	Optional. The alias of a public key in the certificate. If certificateIdentifier is not present, the certificate information is retrieved from the KeyInfo element of signature.

Business Process Examples

The following example business processes illustrate using the XML Digital Signature service:

Example Business Process 1

The following BPML signs the document based on the parameters passed from BPML to the XML Digital Signature service.

```
<process name="xmldsig_enveloped">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="signatureType">enveloped</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

```

<operation name="VerifyMessage">
  <participant name="XMLDSigService"/>
  <output message="verifyRequest">
    <assign to="." from="*" />
    <assign to="action">verify</assign>
    <assign to="certificateIdentifier">test_rsa_pub</assign>
  </output>
  <input message="verifyResponse">
    <assign to="." from="*"></assign>
  </input>
</operation>

</sequence>
</process>

```

Example Business Process 2

The following BPML shows how to sign the Primary Document and add the KeyInfo element within the Signature element by including the certificateIdentifier parameter in the signing request. The example also includes how to construct the Transforms node.

```

<process name="xmldsig_enveloped_keyinfo">
  <sequence>
    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="signatureType">enveloped</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
        <assign to="." from="*" />
        <assign to="action">verify</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>

```

Example Business Process 3

The following BPML shows how to sign particular nodes in the Primary Document. The nodes to be signed are delimited by comma (,). The nodes to be signed include the XML nodes and xpointer path. The signature type is enveloped. This example also includes the construction of the ds:Transforms node.

```
<process name="xmldsig_enveloped_nodetosign_transform">
  <sequence>
    <assign
to="temp/@Algorithm">http://www.w3.org/2000/09/xmldsig#enveloped-signature</assign>
    <assign to="ds:Transforms/ds:Transform" from="temp/@*" />

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="nodeToSign">value1,value2,xpointer(//*[@ID='1'])</assign>
        <assign to="signatureType">enveloped</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="ds:Transforms" from="ds:Transforms/node()" />
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
        <assign to="." from="*" />
        <assign to="action">verify</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

Example Business Process 4

The following BPML shows how to sign particular nodes in the Primary Document. The nodes to be signed are delimited by comma (,). The signature type is enveloping.

```
<process name="xmldsig_enveloping_nodetosign">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*" />
        <assign to="action">sign</assign>
        <assign to="nodeToSign">value1,value2</assign>
```

```

        <assign to="signatureType">enveloping</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
    </output>
    <input message="signResponse">
        <assign to="." from="*"></assign>
    </input>
</operation>

<operation name="VerifyMessage">
    <participant name="XMLDSigService"/>
    <output message="verifyRequest">
        <assign to="." from="*" />
        <assign to="action">verify</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
    </output>
    <input message="verifyResponse">
        <assign to="." from="*"></assign>
    </input>
</operation>

</sequence>
</process>

```

Example Business Process 5

This BPML example shows how to sign particular nodes in the Primary Document, the nodes to be signed are delimited by “,”. The signature type is enveloping.

```

<process name="xmldsig_enveloping_nodetosign">
    <sequence>

        <operation name="SignMessage">
            <participant name="XMLDSigService"/>
            <output message="signRequest">
                <assign to="." from="*" />
                <assign to="action">sign</assign>
                <assign to="nodeToSign">value1,value2</assign>
                <assign to="signatureType">enveloping</assign>
                <assign to="signCertificateIdentifier">test_rsa_priv</assign>
            </output>
            <input message="signResponse">
                <assign to="." from="*"></assign>
            </input>
        </operation>

        <operation name="VerifyMessage">
            <participant name="XMLDSigService"/>
            <output message="verifyRequest">
                <assign to="." from="*" />
                <assign to="action">verify</assign>
                <assign to="certificateIdentifier">test_rsa_pub</assign>
            </output>
            <input message="verifyResponse">
                <assign to="." from="*"></assign>
            </input>
        </operation>
    </sequence>
</process>

```

```
</sequence>
</process>
```

Example Business Process 6

This following input file and BPML shows how to sign multiple documents passed from “documents/document” parameter. The nodes to be signed are delimited by comma (,). The signature type is enveloping.

Input file:

```
<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <nodeToSign>value1,value2</nodeToSign>
  <ds:Transforms xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
      <ds:XPath>descendant-or-self::Contract</ds:XPath>
    </ds:Transform>
  </ds:Transforms>
</document>
<document>
  <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
</document>
</documents>
```

BPML:

```
<process name="xmldsig_enveloping_documents">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">documents</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET">PrimaryDocument</assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
```

```

    <output message="signRequest">
      <assign to="." from="*"></assign>
      <assign to="action">sign</assign>
      <assign to="signatureType">enveloping</assign>
      <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      <assign to="documents" from="documents/node()"></assign>
    </output>
    <input message="signResponse">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="VerifyMessage">
    <participant name="XMLDSigService"/>
    <output message="verifyRequest">
      <assign to="." from="*"></assign>
      <assign to="action">verify</assign>
      <assign to="certificateIdentifier">test_rsa_pub</assign>
    </output>
    <input message="verifyResponse">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

Example Business Process 7

The following input file and BPML shows how to sign multiple documents passed from “documents/document” parameter, Transforms/ds:Transform/ds:XPath is used to sign specific node. The signature type is detached.

Input file:

```

<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <ds:Transforms xmlns:ds="http://www.w3.org/2000/09/xmlsig#">
      <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
        <ds:XPath>descendant-or-self::Contract</ds:XPath>
      </ds:Transform>
    </ds:Transforms>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
  </document>
</documents>

```

BPML:

```

<process name="xmlsig_detached_nodetosign">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
      </output>
    </operation>
  </sequence>
</process>

```



```

    <assign to="mode">xml_to_process_data</assign>
    <assign to="root_element">documents</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation>
  <participant name="ReleaseService"/>
  <output message="releaseRequest">
    <assign to="TARGET">PrimaryDocument</assign>
  </output>
  <input message="releaseResponse">
  </input>
</operation>

<operation name="SignMessage">
  <participant name="XMLDSigService"/>
  <output message="signRequest">
    <assign to="." from="*"></assign>
    <assign to="action">sign</assign>
    <assign to="signatureType">detached</assign>
    <assign to="signCertificateIdentifier">test_rsa_priv</assign>
    <assign to="documents" from="documents/node()"></assign>
  </output>
  <input message="signResponse">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="VerifyMessage">
  <participant name="XMLDSigService"/>
  <output message="verifyRequest">
    <assign to="." from="*"></assign>
    <assign to="action">verify</assign>
    <assign to="certificateIdentifier">test_rsa_pub</assign>
  </output>
  <input message="verifyResponse">
    <assign to="." from="*"></assign>
  </input>
</operation>

</sequence>
</process>

```

Example Business Process 8

The following input file and BPML shows how to sign multiple documents passed from “documents/document” parameter. The signature type is combination.

Note: This example works when the second document is in non-XML format. If the second document is in XML format, remove the ds:Transforms node in the input file below.

Input file:

```
<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <signatureType>enveloped</signatureType>
    <nodeToSign>value1,value2</nodeToSign>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
    <signatureType>enveloping</signatureType>
<ds:Transforms xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64">
    </ds:Transform>
  </ds:Transforms>
</document>
<document>
  <documentID>sgconrad:31e5343c:1158d3b080f:-75bc</documentID>
  <signatureType>detached</signatureType>
  <nodeToSign>node1,node2</nodeToSign>
</document>
</documents>
```

BPML:

```
<process name="xmldsig_combination">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">documents</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET">PrimaryDocument</assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*"></assign>
        <assign to="action">sign</assign>
        <assign to="signatureType">combination</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="documents" from="documents/node()"></assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

    </output>
    <input message="signResponse">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="VerifyMessage">
    <participant name="XMLDSigService"/>
    <output message="verifyRequest">
      <assign to="." from="*"></assign>
      <assign to="action">verify</assign>
      <assign to="certificateIdentifier">test_rsa_pub</assign>
    </output>
    <input message="verifyResponse">
      <assign to="." from="*"></assign>
    </input>
  </operation>

</sequence>
</process>

```

XML Digital Signature Service Examples

The following example signature types illustrate using the XML Digital Signature service:

Example of Enveloped Signature (includes RSA Key Info)

The following example shows the enveloped signature type:

```

<test>
  <value1 ID="1">testval1</value1>
  <value2 ID="2">testval2</value2>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>fgWg+5VNk9ZqUy86IOIE3QS+bGQ=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
MGFS0c419ASo4o00YOI958QFeUG/pY0U3XCa7miUfqw2e5tsZFb7gwE50rvMZe61tyqn1mKh+UKA
20G8r24wNUHuIRM3rtUldrJ3zea4UjpkmfdIqCWGcfDwlnqRTNYY5WDpzJ/JDLUFSIIStv/8VuFH
UR3wZvMecXNv95EqbmA=
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
MIICkDCCAfkCBECDOFYwDQYJKoZIhvcNAQEFBQAwY0xJTAjBgkqhkiG9w0BCQEFnNlcmVuYV9s
aUBzdGVyY292bS5jb20xczAJBgNVBAYTAlNHMREIwEAYDVQQIEWlTaW5nYXBvcmlUeGjAYBgNVBAoT

```

```

EVN0ZXJsaW5nIENvbW1lcmNlMREwDwYDVQQLEWhBc2lhTGFiZcEUMBIGA1UEAxMLEG1sZHNpZ19y
c2EwIBcNMDCxMDAzMDYzODQ2WhgPNzQ4MzA3MjcwNjM4NDZAMIGNMSUwIwYJKoZIhvcNAQkBFhZz
ZXJlbnFfbG1Ac3RlcmNvbW0uY29tMQswCQYDVQGEwJTRzESMBAGA1UECBMJU2luZ2Fwb3JlMR0w
GAYDVQQKEwFTdG9yYyBDb21tZXJjZTERMA8GA1UECwMIQXNpYUxhYnMxMDFDASBgNVBAMTC3ht
bGRzaWdfcnNhMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCqd0H0JBMcJfbuXh5SnKgre9GP
15LMUPJXLFPok5duviihyQ4dpchT7l3cCeQY5VsmxEdsJ04WSXhdzHK9ctY5vHUEM01XR7sYEe8gb
YoQBctP6lkr+RR53fymQBzuvEVVWq8oQXlQ/kJosiSDGoLQ07z0/UDa4/8q9AbeCF0eljQIDAQAB
MA0GCSqGSIb3DQEBAQUAA4GBAJwclB0gbZAr/QdH80xp3Cw2tVosoExORR2GOSwo3Fo4Iq2e4nlq
jQqjIcGha8ZTUicNFaBqVhR8McbMyWqXhD6rjTBX6VpsPmclDQF/HuNcmee22rfhnn7shoHY4c6W
0wiEc/g7MOEQCsslJdoD/gsJTFcP/zYA9r0qMw69Y+rr
</ds:X509Certificate>
  </ds:X509Data>
  <ds:KeyValue>
  <ds:RSAKeyValue>
  <ds:Modulus>
qndB9CQTHCX2714eUpyoK3vRj9eSzfDyVyxaaJOXbr4ockOHaXIU+5d3AnkGOVbJsRHbCdOfk14X
cxyxXLW0bx1BDNNV0e7GBHvIG2KEAXLT+pZEFkUed38pkAZ87hFVVqvKEF5UP5CaLiKgxqC0Du89
P1A2uP/KvQG3ghdHpY0=
</ds:Modulus>
  <ds:Exponent>AQAB</ds:Exponent>
  </ds:RSAKeyValue>
</ds:KeyValue>
</ds:KeyInfo>
  </ds:Signature>
</test>

```

Example of Enveloped Signature with particular nodes (support for xpointer path)

The following example shows the enveloped signature type with particular nodes:

```

<?xml version="1.0" encoding="UTF-8"?>
<test>
  <value1 ID="1">testval1</value1>
  <value2 ID="2">testval2</value2>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
  <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <ds:Reference URI="#1">
  <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <ds:DigestValue>XEZm5p6sjqi+Gt61BHQTOwTL2Nw=</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="#2">
  <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
  </ds:Transforms>
  <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <ds:DigestValue>1k1M9eAUMUWM4FP40A1/YC9qS/Y=</ds:DigestValue>
  </ds:Reference>
  <ds:Reference URI="#xpointer(id('1'))">
  <ds:Transforms>
  <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>

```

```

</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>XEZm5p6sjqi+Gt61BHQTOwTL2Nw=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="#xpointer(//*[@ID='1'])">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>XEZm5p6sjqi+Gt61BHQTOwTL2Nw=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
XEj4pRG/J/xlZEBXTelv92PunsC7/LGKwSWGtQYNDE+nWQLlBGqGi6B7VJ3AYgkHAr+OjZyJWi15
8dQm0Yp+vCK92tDqbnG4ME2ie4WAJwUyI35ZCdU2u/uoBzctAMk4qHTdcZjo97XJw32bR1nFIKb
4XEg52g0+pqnw9Yuccs=
</ds:SignatureValue>
</ds:Signature>
</test>

```

Example of Enveloped Signature with particular nodes, and the Signature node is under to the node to be signed

```

<test>
  <value1 ID="1">testval1<ds:Signature
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="#1">
    <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>XEZm5p6sjqi+Gt61BHQTOwTL2Nw=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>
Q2TuyUoPRJ3bSpdDDegt5AwJSiXxFcgfmyHi8IRa3X/HQpOfocwCLqyHwaJ1/CIgHJ8/iVNkILHF
Z6Aag87PQ5ptdFV0zrnq5RZyF2qzTFJ/r1F2Jjqf0BK1fzHa5H7Sx67qtCD9/ncCia7O3XVd+Sto
JAp/DQVrXw08cSi2RJc=
</ds:SignatureValue>
  </ds:Signature>
</value1>
  <value2 ID="2">testval2<ds:Signature
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="#2">
    <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>

```

```

    <ds:DigestValue>1k1M9eAUMUWM4FP4OA1/YC9qS/Y=</ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
  <ds:SignatureValue>
chRO8fRr9/SmZTrTB3Wh95o7thyRoJ6fX372XWyDHQuCsyC8oFYMUVuJ8fETn1gZ9gSLMh/TNFyx
7VknVNPEK+ZoS0UCw6wbMakVDuV/SHm9Jhi9+K3v2+E9173rVrsBjxIbilfpqRErUxie+uhKWg/M
urKdEfCzHOn7Hw1hKEM=
</ds:SignatureValue>
  </ds:Signature>
</value2>
</test>

```

Example of Enveloping Signature (includes DSA Key Info)

The following example shows the enveloping signature type:

```

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
    <ds:Reference URI="#test">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>cCSTkbrRc0cHJvRfQ4DrxSyqrsM=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>

<ds:SignatureValue>NwAAHYeGf9UUVPeZa+RVd41c9ikwqVhpp9XnzWhq3oqrCt fuxJyFsg==</ds:Sign
atureValue>
  <ds:KeyInfo>
    <ds:X509Data>
      <ds:X509Certificate>
MIICZjCCAiQCBEMAkJAwCwYHKOZIZjgEAWUAMBgxFjAUBGNVBAMTDXRlc3RfZHNhX3ByaXYwIBcN
MDkwMjE3MDMwMTM2WhgPMjI4MjEyMDIwMzAxMzZaMBGxFjAUBGNVBAMTDXRlc3RfZHNhX3ByaXYw
ggG3MIIBLAYHKOZIZjgEATCCAR8CgYEA/X9TgR11EilS30qcLuzk5/YRt1I870QAwX4/gLZRJm1F
XUAiUftZPY1Y+r/F9bow9subVWzXgTuAHTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7V+fG
qKYVDwT7g/bTxR7DAjVUE1oWkTL2dfOuK2HXKu/yIgmZndFIaccCFQCXYFCPFSMLzLKSuYKi64QL
8Fgc9QKBgQD34aCF1ps93su8q1w2uFe5eZsvu/o66oL5V0wLPQeCZ1FZV4661F1P5nEHEIGAtEkW
cSPoTCgWE7fPCTKMyKbhPBZ6i1R8jsjgo64eK7OmdZFuo38L+ie1YvH7YnoBJDvMpPG+qFGQiaid
3+Fa5Z8GkotmXoB7VSVkAUw7/s9JKgOBhAACgYAY5m5jjUnzCSJ92JUv8NAUwZiEbi/znOQpeBdb
rNc6QHTACMBRpdj9CF1YnEF6ShqRrKHAMXiHYnhXEgd34duQ28qMh+Weeq4xHt1ARMnlTIEFrYk1
gIP50wBQ3aK+MR14SAn2OKx59RBTd4E8rI21jYriHP81OXGwQwABqEJs9jALBgcqhkjOOAQDBQAD
LwAwLAIUZRXNX6osaZJBmoSbz5WAq64z+R8CFFRIJ4lor4XV+sM3oohFLziQZtdK
</ds:X509Certificate>
      </ds:X509Data>
    <ds:KeyValue>
      <ds:DSAKeyValue>
        <ds:P>
/X9TgR11EilS30qcLuzk5/YRt1I870QAwX4/gLZRJm1FXUAiUftZPY1Y+r/F9bow9subVWzXgTuA
HTRv8mZgt2uZUKWkn5/oBHsQIsJPu6nX/rfGG/g7V+fGqKYVDwT7g/bTxR7DAjVUE1oWkTL2dfOu
K2HXKu/yIgmZndFIacc=
        </ds:P>
        <ds:Q>l2BQjxUjC8yykrmcouuEC/BYHPU=</ds:Q>
        <ds:G>
9+GghdabPd7LvKtcNrhXuXmUr7v6OuqC+VdMCz0HgmdRWVeOutRZT+ZxBxCBGLRjFnEj6EwoFhO3
zwkyjMim4TwWeotUfI0o4KOUHiuzpnWRbqN/C/ohNWLx+2J6ASQ7zKTxvqhRkImog9/hWuWfBpKL

```

```

Zl6Ae1U1ZAFMO/7PSSo=
</ds:G>
  <ds:Y>
MuZuY41J8wki fdiVL/DQFMGYhGyP85zkKXgXW6zXOkB0wAjAazw4/QhdWJxBekoakayh2jF4h2J4
VxIHd+HbkNvKjIflnnquMR7ZQETJ5bSBBa2JNYCD+dMAUN2ivjEZeEgJ9jisefUQUw+BPKyNtY2K
4hz/JTlxsEMAAhCbPY=
</ds:Y>
  </ds:DSAKeyValue>
  </ds:KeyValue>
  </ds:KeyInfo>
  <ds:Object Id="test">
  <test>
  <value1 ID="1">testval1</value1>
  <value2 ID="2">testval2</value2>
  </test>
  </ds:Object>
</ds:Signature>

```

Example of Enveloping Signature with particular nodes

The following example shows the enveloping signature type with particular nodes:

```

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="#value1">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>v5xgwt+7FN8Y7L7FeUX9ywugZZA=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#value2">
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>c/+kr7SZTA7LuQoR/5QEpktn5I=</ds:DigestValue>
    </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
KXhmImkXxL315S/yLqN6U/FlfcUUVC5s8KPyR8zTQaJ3feZclgd1PGCx3ewZGLbJwT6GM1YY6mXL
7uL4Ylrxjo4+/gzfTKwjILTq1UKx1/mrwgXQ3VV+mgQi2neA2Mvo0dAx9T1lm+HsphsMZhkNKtu
AwRwMtbbx3LszgzUFhE=
    </ds:SignatureValue>
    <ds:Object Id="value1">
    <value1 ID="1">testval1</value1>
    </ds:Object>
    <ds:Object Id="value2">
    <value2 ID="2">testval2</value2>
    </ds:Object>
  </ds:Signature>

```

Example of Enveloping Signature with non-XML Input File

The following example shows the enveloping signature type with non-XML input file:

```

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>

```



```

0wiEc/g7MOEQCss1JdoD/gsjTfCP/zYA9r0qMw69Y+rr
</ds:X509Certificate>
  </ds:X509Data>
  <ds:KeyValue>
  <ds:RSAKeyValue>
  <ds:Modulus>
qndB9CQTHCX2714eUpyoK3vRj9eSzFDyVyxaaJOXbr4ockOHaXIU+5d3AnkGOVbJsRHbCdOFk14X
cxyvXLW0bx1BDNNV0e7GBHVIG2KEAXLT+pZEfkUed38pkAZ87hFVVqvKEF5UP5CaLIkgxqC0Du89
P1A2uP/KvQG3ghdHpY0=
</ds:Modulus>
  <ds:Exponent>AQAB</ds:Exponent>
  </ds:RSAKeyValue>
  </ds:KeyValue>
  </ds:KeyInfo>
</ds:Signature>

```

Example of Detached Signature with particular nodes

```

<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:SignedInfo>
    <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
    <ds:Reference URI="sgritz:node1:11fccec1123:8071">
    <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
    <ds:XPath>descendant-or-self::value1</ds:XPath>
    </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>G8NrCW6+DmK8I9129pEmtV7Yfr0=</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="sgritz:node1:11fccec1123:8071">
    <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
    <ds:XPath>descendant-or-self::value2</ds:XPath>
    </ds:Transform>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>YTD9jkgFYn1ABR2ywRH16aphIZzk=</ds:DigestValue>
    </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>
cq2Rxc5gQtgAakjKQ2l+nrMgffqVuEK1RYtV3qFhSPzWLe4YnFbmM15JziNBkCa7A4Kxzm3ragh
ikyTja9YJrRkSlIeN7dUev67TTFMyzwd0Tk0159sVBhBgabGhFYbH4myiW0mPdwBajGcc1Mg2wDI
N50no5Tb/L3RAJh8BiA=
</ds:SignatureValue>
</ds:Signature>

```

Example of Combination Signature of Enveloped/Enveloping/Detached

The following example shows the combination signature type:

```

<test>
  <value1 ID="1">testval1</value1>
  <value2 ID="2">testval2</value2>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">

```

```

    <ds:SignedInfo>
      <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="">
      <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>fgWg+5VNk9ZqUy86IOIE3QS+bGQ=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="sgritz:node1:11fccec1123:8182">
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>+iuNHAvWu/xsJrDy9GVarpQH3HA=</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#sgritz:node1:11fccec1123:8200">
      <ds:Transforms>
      <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64"/>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>a3yRsN5knCCHB751ql7pXdbcNfo=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
FHXtRjrgkxZZTw+ryAzH28WtanwpHvo0KaeJTe08h9g/EHNvWbt0uJcUIH43oqizl3zqEqHdhIFF
CGrc6UBt7QVMh245PJe73cHDwgGBWpVRT08y34wEPaxz+mY46WIlb4g3qrgYMqyZotjKFVRU9/4e
Ii+DpfbW9qUvPF7zVSI=
</ds:SignatureValue>
      <ds:Object Encoding="Base64" Id="sgritz:node1:11fccec1123:8200">
dGhpcyBpcyBhbiBub24teG1sIHRLc3QgZG9jIGZvciB4bWwgZHNpZyBzZXJ2aWNlLgo=
</ds:Object>
    </ds:Signature>
</test>

```

XML Encoder Service

Note: If the input document character encoding is specified in Application, it overrides the encoding specified in the map. The output document content type and character encoding are set according to the information in the map.

The following table provides an overview of the XML Encoder service:

System name	XMLEncoderType
Graphical Process Modeler (GPM) categories	All Services, Translation
Description	Translates the primary document into XML using a specified map, replacing the primary document with the result of the translation. The XML Encoder service is used within business processes to translate of the primary document into XML.
Preconfigured?	An instance of this service is created upon installation but is not configured nor is any configuration required other than specifying values for the parameters when used within a business process.
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	The map specified in the map_name parameter must have been registered with Application and activated. If either of these conditions is not met then the translation will not be performed.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	<ul style="list-style-type: none">◆ Success – Translation was successful.◆ Error – Errors were encountered during translation or translation could not be performed. The Translator report contained in the Workflow Context Status report should be consulted for further detail.
Restrictions	No
Persistence level	None
Testing considerations	The best way to test is within a simple business process where the XMLEncoder service is the only operation. After execution of the business process the output can be verified Application and the translator report can be viewed for detail on what occurred during the translation.

Implementing the XML Encoder Service

To implement the XML Encoder service, complete the following tasks:

1. Activate your license for the XML Encoder service. See *An Overview of Implementing Services*.
2. Create an XML Encoder service configuration. See *Creating a Service Configuration*.
3. Configure the XML Encoder service. See *Configuring the XML Encoder Service* on page 22.
4. Use the XML Encoder service in a business process.

Configuring the XML Encoder Service

To configure the XML Encoder service, you must specify settings for the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
edi_input_decimal_separator	Character used to indicate the decimal point on the input side.
edi_input_element_delimiter	Character used to delimit elements (fields) on the input side.
edi_input_release_character	Character used to quote elements (fields) that contain the delimiter on the input side.
edi_input_repeating_element_delimiter	Character used to delimit repeating elements on the input side.
edi_input_segment_delimiter	Character used to delimit segments on the input side.
edi_input_sub_element_delimiter	Character used to delimit sub-elements on the input side.
edi_input_tag_delimiter	Character used to delimit tags on the input side.
exhaust_input	Whether to execute the map until the Translation service has translated all of the input. Valid values are Yes and No.
map_name	Used as the map for translation. Select the map from the list of registered maps in Application. Only registered maps can be used.
mode	Whether translation is performed on an input file. If the input file is already in XML format, then the file can be imported directly attached to the process data. The file can then be pulled from the process data and written to a document in the file system. Valid values are: <ul style="list-style-type: none">◆ Encode non-XML document◆ Use existing XML document◆ Create document using XPATH
output_to_process_data	Whether the output of the translation should be placed in the process data tree. Valid values are Yes and No (default).
root_element	Root element name of the document containing result of XPath expression when value in Mode field is process_data_to_document mode. Value is a string.

Field	Description
XPath	XPath when value in Mode field is process_data_to_document mode. Value is a string.

XML Encryption Service

The following table provides an overview of the XML Encryption service:

System name	XMLEncryption
Graphical Process Modeler (GPM) categories	All Services, Translation
Description	Invokes the Business Service Handler to interpret and execute activities defined in the BPSS.
Business usage	<ul style="list-style-type: none">◆ Encrypts primary document or specified document using the document ID◆ Decrypts primary document or specified document using the document ID.
Usage example	You have a business process that needs to XML-encrypt a document as part of the security requirements.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Not applicable
Business process context considerations	No
Returned status values	None
Restrictions	No
Persistence level	System default
Testing considerations	None

Output from Service to Business Process

The following table describes the output from the XML Encryption service to the business process:

Parameter	Description
wfc	Encrypted/Decrypted document will be stored in the Primary Doc if no incomingDoc value was specified. Optional.
encryptedEnvelope/incomingDoc	The document ID for the encrypted/decrypted doc will be stored in a valid ebXML node structure if the incomingDoc node was specified. Optional.

Output from Business Process to Service

The following table describes the output from the business process to the XML Encryption service:

Parameter	Description
incomingDoc	The ebXML node structure containing the multiple documents. Valid value is valid ebXML node structure containing the document ids. If not entered, primary document will be encrypted/decrypted. Optional.
alias	The alias name of the public/private key used in encryption/decryption. Required.
encMode	The mode of operation. Valid values are ENCRYPT and DECRYPT. Required.
encType	The type of operation. Currently, only DOCUMENT is supported. Valid value is DOCUMENT. Required.

Output Parameters

This section contains examples of using output parameters for the XML Encryption service.

Output Parameters – Encryption of multiple documents

Encryption of multiple documents can only be done if the input node follows the ebXML MIME node structure. The following is an example of how you would invoke the service:

```
<operation name="ValidateDoc">
  <participant name="XMLEncryption"/>
  <output message="validateDoc">
    <assign to="incomingDoc" from="ebXMLMessage/mime:message"/>
    <assign to="alias" from="string(cpa/tp:CollaborationProtocolAgreement/tp:
PartyInfo[1]/tp:Certificate[@tp:certId=//EncryptCertId/text()/ds:KeyInfo/ds:KeyName
)"/>
    <assign to="encMode" from="'ENCRYPT'"/>
    <assign to="encType" from="'DOCUMENT'"/>
  </output>
  <input message="validateDocResponse">
    <assign to="ebXMLMessage" from="encryptedEnvelope/incomingDoc/node()"/>
  </input>
</operation>
```

Output Parameters – Encryption using the Primary Document

The following example illustrates how to invoke the service for encryption using the primary document:

```
<operation name="ValidateDoc">
  <participant name="XMLEncryption"/>
  <output message="validateDoc">
    <assign to="alias" from="'SomePublicKeyName'"/>
    <assign to="encMode" from="'ENCRYPT'"/>
    <assign to="encType" from="'DOCUMENT'"/>
  </output>
  <input message="validateDocResponse">
    <assign to="." from="*" />
  </input>
</operation>
```

Output Parameters – Decryption of Multiple Documents

Decryption of multiple documents can only be done if the input node follows the ebXML mime node structure. The following example shows how you would invoke the service:

```
<operation name="ValidateDoc">
  <participant name="XMLEncryption"/>
  <output message="validateDoc">
    <assign to="incomingDoc" from="InboundMIME/mime:message"/>
    <assign to="alias"
from="string(cpa/tp:CollaborationProtocolAgreement/tp:PartyInfo[1]/tp:Certificate[@t
p:certId = //DecryptCertId/text()/ds:KeyInfo/ds:KeyName)"/>
    <assign to="encMode" from="'DECRYPT'"/>
    <assign to="encType" from="'DOCUMENT'"/>
  </output>
  <input message="validateDocResponse">
    <assign to="InboundMIME" from="encryptedEnvelope/incomingDoc/node()"/>
  </input>
</operation>
```

Output Parameters – Decryption using the Primary Document

For decryption using the primary document, the following is an example of how you would invoke the service:

```
<operation name="ValidateDoc">
  <participant name="XMLEncryption"/>
  <output message="validateDoc">
    <assign to="alias" from="'SomePrivateKeyName'"/>
    <assign to="encMode" from="'DECRYPT'"/>
    <assign to="encType" from="'DOCUMENT'"/>
  </output>
  <input message="validateDocResponse">
    <assign to="." from="*" />
  </input>
</operation>
```


Implementing the XML Encryption Service

To implement the XML Encryption service for use in a business process:

1. Create an XML Encryption service configuration. For information, see *Managing Services and Adapters*.
2. Use the XML Encryption service in a business process.

XML Validation Service

The following table provides an overview of the XML Validation service:

System name	XMLValidator WellFormed
Graphical Process Modeler (GPM) categories	All Services, Translation
Description	Validates XML documents in process data. It can validate against a schema or a DTD, or verify that the document is well-formed.
Business usage	Used within business processes to perform validation of a XML document passed to the business process or generated within the business process.
Usage example	<p>You have a business process that performs the following:</p> <ul style="list-style-type: none">◆ Translates an incoming EDI document◆ Converts the translated document to XML format◆ Feeds the XML document to an enterprise application <p>You could use the XML Validation service to validate the format of the XML document before feeding it to the enterprise application.</p>
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	No
Application requirements	<p>If the XML document will be verified using a DTD or schema, then you must specify one of the following in the XML document:</p> <ul style="list-style-type: none">◆ A reference to the location of the DTD or schema◆ An actual DTD <p>If you specify a reference to the DTD or schema location, the DTD or schema must be checked in to Application.</p>
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	Uses the primary document and can validate documents in process data.
Returned status values	<ul style="list-style-type: none">◆ Success – Transformation was successful.◆ Error – Errors were encountered during transformation or transformation could not be performed. <p>See the Workflow Context Status report for further details.</p>

Restrictions	No
Persistence level	None
Testing considerations	None

Requirements

The DTDs and schemas are checked in Application. If you want to validate using a schema or DTD, you must specify either the location and name of the schema or DTD, or the actual DTD, in the XML document (which is the primary document of the BPML).

How the XML Validation Service Works

If you have a business process that uses the translation service to create XML, you can use the XML Validation service to check the output of that service to verify its credibility before operating on it.

Business Process Example

The illustration shows an example business process that uses the XML Validation service:

```
<process name="XMLVal_WellFrmd">
  <sequence>
    <operation name="Validate">
      <participant name="XMLValidator"/>
      <output message="outputMessage">
        <assign to="." from="*"></assign>
        <assign to ="xml_input_validation">WELLFORMED</assign>
      </output>
      <input message="inputMessage">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Implementing the XML Validation Service

To implement the XML Validation service, complete the following tasks:

1. Activate your license for the XML Validation service. See *An Overview of Implementing Services*.
2. Create an XML Validation service configuration. See *Creating a Service Configuration*.
3. Configure the XML Validation service. See *Configuring the XML Validation Service* on page 1724.
4. Check in any DTDs or schemas that will be used to validate the XML documents.
5. Use the XML Validation service in a business process.

Configuring the XML Validation Service

To configure the XML Validation service, you must specify field settings in Application and using BPML. See *Creating a Service Configuration*.

Application Configuration

The following table describes the fields used to configure the XML Validation service in Application:

Field	Description
Name	Unique and meaningful name for the service configuration. Required.
Description	Meaningful description for the service configuration, for reference purposes. Required.
Select a Group	Select one of the options: <ul style="list-style-type: none">◆ None – You do not want to include this configuration in a group at this time.◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. Note: See <i>Using Service Groups</i> .
Input Validation (xml_input_validation)	Specifies the method used for validation of the input XML file. Required. Valid values are: <ul style="list-style-type: none">◆ Validate well-formed – Validates that the input file is a well-formed XML document◆ Validate using a DTD – Validates the input file against a DTD◆ Validate using an XML schema – Validates the input file against an XML schema Note: This parameter can be overwritten using BPML.

BPML Configuration

The following table describes the fields used to configure the XML Validation service using BPML:

Field	Description
xml_input_validation	Specifies the method used for validation of the input XML file. Required. Valid values are: <ul style="list-style-type: none">◆ WELLFORMED – Validates that the input file is a well-formed XML document◆ DTD – Validates the input file against a DTD◆ SCHEMA – Validates the input file against an XML schema Note: This parameter is set in the Application configuration, but can be overwritten using BPML.
xml_input_from	Specifies the location of the XML to validate. Optional. Valid values: <ul style="list-style-type: none">◆ Procdata – Validate the XML in process data◆ Primdoc – Validate the XML in the primary document Default is Primdoc.

Field	Description
input_pd_xpath	Identifies the XPath location of the XML to validate. Required if xml_input_from is set to Procddata .

XSLT Service (Build 4300 - Build 4320)

The following table provides an overview of the XSLT service:

System name	XSLT Service
Graphical Process Modeler (GPM) categories	All Services, Translation, Sync Mode, Transactional Mode
Description	Enables you to use XSLT style sheets in Application. The XSLT service performs transformation of an XML document from specified location (primary document or process data) using selected XSLT. It can also do input XML validation.
Business usage	Could be used to perform any sort of transformation on XML documents where the output is specified in the XSLT itself. The service could be used to produce static HTML page using data from input XML.
Usage example	There is an automotive parts ordering service, where the manufacturer receives an XML document (containing an order) from a supplier. The manufacturer can set up a business process that uses this service to transform the XML to another format that their system can understand.
Preconfigured?	Yes
Requires third party files?	You may need to check in XSLT stylesheets.
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	The service uses and modifies the business process context document content.
Returned status values	<ul style="list-style-type: none">◆ Basic status Success – Transformation was successful.◆ Basic status Error – Errors were encountered during transformation or transformation could not be performed. See the report contained in the business process context status report for further detail.
Restrictions	None
Persistence level	None
Testing considerations	Problems to look for would be malformed or invalid XML and XSLT. Verify valid output of the transformation. If the transformer fails to allocate a field in XML data during transformation, it does not report it as an error; instead, leaves it blank.

Requirements

Before you configure the XSLT service in the GPM, you must:

Be proficient in XSLT.

Check in any XSLT style sheets you want to use. See *Checking In an XSLT Style Sheet*.

Implementing the XSLT Service

To implement the XSLT service, complete the following tasks:

1. Activate your license for the XSLT service.
2. Create an XSLT service configuration, if necessary. Application provides a standard configuration of the XSLT service for you (named XSLTService). You do not need to create one. However, you may choose to create a unique XSLT service configuration.
3. Configure the XSLT service.
4. Check in any XSLT style sheets. See *Managing XSLT Style Sheets*.
5. Use the XSLT service in a business process.

Configuring the XSLT Service

Application provides a standard configuration of the XSLT service for you (named XSLTService). You do not need to create one. However, you may choose to create a unique XSLT service configuration.

To configure the XSLT service, you must configure the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
additional_xslt_parms	Where you specify additional parameters in the XSLT style sheet. Click this field, then the ellipses to enter key-value pairs in the Value of additional_xslt_parms dialog box. Click the icon button to the right to use the XPath Expression Builder.
input_pd_xpath	Location of the input XML in the process data document using XPath, if the XML document comes from process data. Required if the value for xml_input_from is process data. Click this field, then the ellipses to enter key-value pairs in the Value of input_pd_xpath dialog box. Click the icon button to the right to use the XPath Expression Builder.
xml_input_from	Where the service should receive the XML document from, either primary document or process data.
xml_input_validation	Select No validation if you do not want to validate the input XML document or select dtd or schema to use either one to validate the input XML document.
xslt_name	Previously checked in XSLT style sheet that you want to use.

Field	Description
load_from_classpath	If set to true, the system will look for the template (named by the xslt_name parameter) on the classpath. Valid values are true and false. Default is false. Optional.

Managing XSLT Style Sheets

Managing XSLT style sheets involves the following tasks:

- Checking In an XSLT Style Sheet
- Checking In an XSLT Style Sheet Using the Text Editor
- Searching for an XSLT Style Sheet
- About Search Results
- Editing an XSLT Style Sheet
- Checking In an Updated Version of an XSLT Style Sheet
- Checking Out an XSLT Style Sheet
- Enabling or Disabling an XSLT Style Sheet
- Specifying a Default XSLT Style Sheet

Checking In an XSLT Style Sheet

To use XSLT style sheets in Application, you must first check them in.

To check in an XSLT style sheet to Application:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Check-in**, click **Go!**
3. Type the name of the XSLT style sheet.
4. For the input mode, select **Check-in style sheet** and click **Next**.
5. For the XSLT Style Sheet filename, type the path to the XSLT style sheet or click **Browse**, locate the style sheet on your local disk, and click **Open**. The name should not have spaces or apostrophes in it.
6. Type comments in the **Check-in Comments** field.

Tip: Use the Check-in Comments field to note the purpose of the XSLT style sheet or explain the changes made to it.
7. Select the encoding that most closely matches the style sheet encoding and click **Next**.
8. If you do not want the XSLT style sheet to be enabled, click the **Enable for Business Processes** check box to clear it.
9. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes.
 - ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking In an XSLT Style Sheet Using the Text Editor

You can also check in XSLT style sheets by typing or copying the content of an XSLT into the text editor.

To check an XSLT style sheet in to Application using the text editor:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Check-in**, click **Go!**
3. Type the name of the XSLT style sheet.
4. For the input mode, select **Style Sheet Text Editor** and click **Next**.
5. Type a description of the style sheet.
6. Under **XSL Style Sheet**, type or copy the content of the style sheet and click **Next**.

Note: The text editor does not validate the style sheet.

7. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes.
 - ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Searching for an XSLT Style Sheet

To check in a new version, check out, enable, or disable an XSLT style sheet, you must first specify which one you want. You can locate an XSLT by name or from an alphabetic list.

Searching by name is more precise and provides fewer results. Searching from an alphabetical list shows all XSLT style sheets or ones beginning with a specified letter or digit.

To search for an XSLT style sheet by name:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Search**, type the name of the XSLT style sheet. Case does not matter and you can type part of a name and click **Go!**

Application returns a list of matches unless no XSLT style sheets meet your criteria.

To search for an XSLT style sheet from a list:

1. From the **Deployment** menu, select **XSLT**.
2. Under **List**, select **All** or a specific letter or digit (0 - 9) and click **Go!**

Application returns a list of matches unless no XSLT style sheets meet your criteria.

About Search Results

When you search for an XSLT style sheet, Application returns a results page. The results are displayed in a three-column table. Each row contains icons for the Source Manager and the Version Manager, the XSLT name, and XSLT type. You can sort the list alphabetically by name or type.

Source Manager

The Source Manager enables you to check out an XSLT style sheet and check in a new version of that style sheet. It also displays the following information about an XSLT style sheet:

Date that the XSLT style sheet was checked in
Name of the user who checked in the XSLT style sheet
Comments about changes that have been made

Version Manager

The Version Manager enables you to enable or disable a version of an XSLT style sheet. If there are two or more versions, you can select a default.

The Version Manager also displays the following information about an XSLT style sheet and any of its versions:

Which version is the default version
Date that the XSLT style sheet version was checked in
Name of the user who checked in the XSLT style sheet version
Comments about changes that have been made

Editing an XSLT Style Sheet

After you have checked in a style sheet to Application, you can edit it without checking it out of Application.

To edit an XSLT style sheet in Application:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to edit. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet you want to edit, click **source manager**.
4. Next to the version you want to edit, click **edit**.
5. Type a description of the changes you want to make to the style sheet.
6. Under **XSLT Style Sheet**, edit the style sheet as necessary and click **Next**.

Note: The text editor does not validate the style sheet.

7. Select which version you want to be the default and click **Next**.
8. Review the settings for the XSLT style sheet. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes.
 - ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking In an Updated Version of an XSLT Style Sheet

If you update an XSLT style sheet that has been checked in to Application, you need to check in that style sheet again as an updated version.

To check an updated version of an XSLT style sheet in to Application:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet for which you want to check in a new version. For more information, see *Searching for an XSLT Style Sheet*.

3. Next to the XSLT style sheet for which you want to check in a new version, click **source manager**.
4. Next to **Check-in** a new version of this XSLT style sheet, click **Go!**
5. Type the path to the XSLT style sheet or click **Browse**, locate the XSLT style sheet, and click **Open**.
6. Type comments in the **Check-in comments** field and click **Next**. This field is required.
Tip: Use the Check-in comments field to note the purpose of the XSLT style sheet or explain the changes made to it.
7. Select the version you want to be the default and click **Next**.
8. If you do not want the XSLT style sheet to be enabled, click the **Enable for Business Processes** check box to clear it.
9. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes. Application displays the message, *The system update has completed successfully*.
 - ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking Out an XSLT Style Sheet

To edit an XSLT style sheet that has been checked in to Application and prevent anyone from modifying the file while you are making changes, you check out a version from Application. Checking out locks the source XSLT style sheet so that no one else can edit it while you are editing it. Use the Source Manager to check out a version of an XSLT style sheet.

To check out a version of an XSLT style sheet from Application:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to check out. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet you want to check out, click **source manager**.
4. Next to the version you want to check out, select the encoding.
Note: If a version has been checked in with an encoding other than the Application default of UTF-8, then you can check it out in UTF-8 or any other encoding that the style sheet has been checked in with.
5. Click **check-out**.
6. Select **Save** then click **OK**. Application prompts you to choose a destination location. Browse to the location and click **OK** to save the file and complete checkout.

Enabling or Disabling an XSLT Style Sheet

Enabling a XSLT style sheet makes it available to the Application services and business processes.

You can enable or disable an XSLT style sheet in two ways:

At the time you check it in to Application

Through the Version Manager after the style sheet has been checked in

To enable or disable an XSLT style sheet with the Version Manager:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to enable or disable.
3. Next to the XSLT style sheet you want to enable or disable, click **version manager**.
 - ◆ To enable an XSLT style sheet, click the empty **Enable** box and click **Save**. A check mark indicates the XSLT style sheet is enabled.
 - ◆ To disable an XSLT style sheet, click the checked **Enable** box and click **Save**. An empty box indicates the XSLT style sheet is disabled.

Specifying a Default XSLT Style Sheet

The default XSLT style sheet is the version that is available to business processes. One version must be selected as the default.

To specify a default XSLT style sheet:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want and click **version manager**.
3. Select the version you want to be the default and click **Save**.

XSLT Service (Build 4321 or higher)

The following table provides an overview of the XSLT service:

System name	XSLT Service
Graphical Process Modeler (GPM) categories	All Services, Translation, Sync Mode, Transactional Mode
Description	Enables you to use XSLT style sheets in Application. The XSLT service performs transformation of an XML document from specified location (primary document or process data) using selected XSLT. It can also do input XML validation.
Business usage	Could be used to perform any sort of transformation on XML documents where the output is specified in the XSLT itself. The service could be used to produce static HTML page using data from input XML.
Usage example	There is an automotive parts ordering service, where the manufacturer receives an XML document (containing an order) from a supplier. The manufacturer can set up a business process that uses this service to transform the XML to another format that their system can understand.
Preconfigured?	Yes
Requires third party files?	You may need to check in XSLT stylesheets.
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	The service uses and modifies the business process context document content.
Returned status values	<ul style="list-style-type: none">◆ Basic status Success – Transformation was successful.◆ Basic status Error – Errors were encountered during transformation or transformation could not be performed. See the report contained in the business process context status report for further detail.
Restrictions	None
Persistence level	None
Testing considerations	Problems to look for would be malformed or invalid XML and XSLT. Verify valid output of the transformation. If the transformer fails to allocate a field in XML data during transformation, it does not report it as an error; instead, leaves it blank.

Requirements

Before you configure the XSLT service in the GPM, you must:

Be proficient in XSLT.

Check in any XSLT style sheets you want to use. See *Checking In an XSLT Style Sheet*.

Memory Requirements

To process large files, the physical memory allocated to Application must be at least six times the size of the file to be processed. See *Performance Tuning Utility* to allocate more physical memory to Application.

The physical memory allocated to Application divided by a memory factor indicates the maximum file size that can be processed by the service. However, it depends on the load of the machine at the time of processing.

Let us consider the following example to understand the concept of memory to be allocated to Application and the memory factor to process a file.

Memory Allocated to Application	Memory Factor	Approximate File Size that can be Processed (Memory Allocated to Application / Memory Factor)
<= 1024 MB	8	Assuming that memory allocated to Application is 1024 MB, the XSLT service can process a 128 MB (1024 / 8) file.
> 1024 MB and <= 1536 MB	7	Assuming that memory allocated to Application is 1280 MB, the XSLT service can process a 182 MB (1280 / 7) file.
> 1536 MB	6	Assuming that memory allocated to Application is 1536 MB, the XSLT service can process a 256 MB (1536 / 6) file.

Implementing the XSLT Service

To implement the XSLT service, complete the following tasks:

1. Activate your license for the XSLT service.
2. Create an XSLT service configuration, if necessary. Application provides a standard configuration of the XSLT service for you (named XSLTService). You do not need to create one. However, you may choose to create a unique XSLT service configuration.
3. Configure the XSLT service.
4. Check in any XSLT style sheets. See *Managing XSLT Style Sheets*.
5. Use the XSLT service in a business process.

Configuring the XSLT Service

Application provides a standard configuration of the XSLT service for you (named XSLTService). You do not need to create one. However, you may choose to create a unique XSLT service configuration.

To configure the XSLT service, you must configure the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
additional_xslt_parms	Where you specify additional parameters in the XSLT style sheet. Click this field, then the ellipses to enter key-value pairs in the Value of additional_xslt_parms dialog box. Click the icon button to the right to use the XPath Expression Builder.
input_pd_xpath	Location of the input XML in the process data document using XPath, if the XML document comes from process data. Required if the value for xml_input_from is process data. Click this field, then the ellipses to enter key-value pairs in the Value of input_pd_xpath dialog box. Click the icon button to the right to use the XPath Expression Builder.
xml_input_from	Where the service should receive the XML document from, either primary document or process data.
xml_input_validation	Select No validation if you do not want to validate the input XML document or select dtd or schema to use either one to validate the input XML document.
xslt_name	Previously checked in XSLT style sheet that you want to use.
load_from_classpath	If set to true, the system will look for the template (named by the xslt_name parameter) on the classpath. Valid values are true and false. Default is false. Optional.
incremental_transform	A performance feature of the transformer. If set to true and physical memory allocated to Application increased, XSLT service can process large files according to the memory requirements. Valid values are True and False. Default is False. Optional.

Managing XSLT Style Sheets

Managing XSLT style sheets involves the following tasks:

- Checking In an XSLT Style Sheet
- Checking In an XSLT Style Sheet Using the Text Editor
- Searching for an XSLT Style Sheet
- About Search Results
- Editing an XSLT Style Sheet
- Checking In an Updated Version of an XSLT Style Sheet
- Checking Out an XSLT Style Sheet

Enabling or Disabling an XSLT Style Sheet
Specifying a Default XSLT Style Sheet

Checking In an XSLT Style Sheet

To use XSLT style sheets in Application, you must first check them in.

To check in an XSLT style sheet to Application:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Check-in**, click **Go!**
3. Type the name of the XSLT style sheet.
4. For the input mode, select **Check-in style sheet** and click **Next**.
5. For the XSLT Style Sheet filename, type the path to the XSLT style sheet or click **Browse**, locate the style sheet on your local disk, and click **Open**. The name should not have spaces or apostrophes in it.
6. Type comments in the **Check-in Comments** field.
Tip: Use the Check-in Comments field to note the purpose of the XSLT style sheet or explain the changes made to it.
7. Select the encoding that most closely matches the style sheet encoding and click **Next**.
8. If you do not want the XSLT style sheet to be enabled, click the **Enable for Business Processes** check box to clear it.
9. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes.
 - ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking In an XSLT Style Sheet Using the Text Editor

You can also check in XSLT style sheets by typing or copying the content of an XSLT into the text editor.

To check an XSLT style sheet in to Application using the text editor:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Check-in**, click **Go!**
3. Type the name of the XSLT style sheet.
4. For the input mode, select **Style Sheet Text Editor** and click **Next**.
5. Type a description of the style sheet.
6. Under **XSL Style Sheet**, type or copy the content of the style sheet and click **Next**.
Note: The text editor does not validate the style sheet.
7. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes.

- ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Searching for an XSLT Style Sheet

To check in a new version, check out, enable, or disable an XSLT style sheet, you must first specify which one you want. You can locate an XSLT by name or from an alphabetic list.

Searching by name is more precise and provides fewer results. Searching from an alphabetical list shows all XSLT style sheets or ones beginning with a specified letter or digit.

To search for an XSLT style sheet by name:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Search**, type the name of the XSLT style sheet. Case does not matter and you can type part of a name and click **Go!**

Application returns a list of matches unless no XSLT style sheets meet your criteria.

To search for an XSLT style sheet from a list:

1. From the **Deployment** menu, select **XSLT**.
2. Under **List**, select **All** or a specific letter or digit (0 - 9) and click **Go!**

Application returns a list of matches unless no XSLT style sheets meet your criteria.

About Search Results

When you search for an XSLT style sheet, Application returns a results page. The results are displayed in a three-column table. Each row contains icons for the Source Manager and the Version Manager, the XSLT name, and XSLT type. You can sort the list alphabetically by name or type.

Source Manager

The Source Manager enables you to check out an XSLT style sheet and check in a new version of that style sheet. It also displays the following information about an XSLT style sheet:

- Date that the XSLT style sheet was checked in
- Name of the user who checked in the XSLT style sheet
- Comments about changes that have been made

Version Manager

The Version Manager enables you to enable or disable a version of an XSLT style sheet. If there are two or more versions, you can select a default.

The Version Manager also displays the following information about an XSLT style sheet and any of its versions:

- Which version is the default version
- Date that the XSLT style sheet version was checked in
- Name of the user who checked in the XSLT style sheet version
- Comments about changes that have been made

Editing an XSLT Style Sheet

After you have checked in a style sheet to Application, you can edit it without checking it out of Application.

To edit an XSLT style sheet in Application:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to edit. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet you want to edit, click **source manager**.
4. Next to the version you want to edit, click **edit**.
5. Type a description of the changes you want to make to the style sheet.
6. Under **XSLT Style Sheet**, edit the style sheet as necessary and click **Next**.

Note: The text editor does not validate the style sheet.

7. Select which version you want to be the default and click **Next**.
8. Review the settings for the XSLT style sheet. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes.
 - ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking In an Updated Version of an XSLT Style Sheet

If you update an XSLT style sheet that has been checked in to Application, you need to check in that style sheet again as an updated version.

To check an updated version of an XSLT style sheet in to Application:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet for which you want to check in a new version. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet for which you want to check in a new version, click **source manager**.
4. Next to **Check-in** a new version of this XSLT style sheet, click **Go!**
5. Type the path to the XSLT style sheet or click **Browse**, locate the XSLT style sheet, and click **Open**.
6. Type comments in the **Check-in comments** field and click **Next**. This field is required.

Tip: Use the Check-in comments field to note the purpose of the XSLT style sheet or explain the changes made to it.

7. Select the version you want to be the default and click **Next**.
8. If you do not want the XSLT style sheet to be enabled, click the **Enable for Business Processes** check box to clear it.
9. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
 - ◆ If Yes, click **Finish** to apply your changes. Application displays the message, *The system update has completed successfully*.

- ◆ If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking Out an XSLT Style Sheet

To edit an XSLT style sheet that has been checked in to Application and prevent anyone from modifying the file while you are making changes, you check out a version from Application. Checking out locks the source XSLT style sheet so that no one else can edit it while you are editing it. Use the Source Manager to check out a version of an XSLT style sheet.

To check out a version of an XSLT style sheet from Application:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to check out. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet you want to check out, click **source manager**.
4. Next to the version you want to check out, select the encoding.
Note: If a version has been checked in with an encoding other than the Application default of UTF-8, then you can check it out in UTF-8 or any other encoding that the style sheet has been checked in with.
5. Click **check-out**.
6. Select **Save** then click **OK**. Application prompts you to choose a destination location. Browse to the location and click **OK** to save the file and complete checkout.

Enabling or Disabling an XSLT Style Sheet

Enabling a XSLT style sheet makes it available to the Application services and business processes.

You can enable or disable an XSLT style sheet in two ways:

At the time you check it in to Application

Through the Version Manager after the style sheet has been checked in

To enable or disable an XSLT style sheet with the Version Manager:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to enable or disable.
3. Next to the XSLT style sheet you want to enable or disable, click **version manager**.
 - ◆ To enable an XSLT style sheet, click the empty **Enable** box and click **Save**. A check mark indicates the XSLT style sheet is enabled.
 - ◆ To disable an XSLT style sheet, click the checked **Enable** box and click **Save**. An empty box indicates the XSLT style sheet is disabled.

Specifying a Default XSLT Style Sheet

The default XSLT style sheet is the version that is available to business processes. One version must be selected as the default.

To specify a default XSLT style sheet:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want and click **version manager**.
3. Select the version you want to be the default and click **Save**.

Yantra Adapters (Build 4300 - Build 4311)

The Yantra adapters enable you to use Yantra functionality from within Gentran Integration Suite. The following table provides an overview of the Yantra adapters:

System Name	These adapters are dynamically created when Gentran Integration Suite is started and the names are in the form of Yantra <API group>. For example, Yantra Inventory, Yantra Order.
Graphical Process Modeler (GPM) categories	All Services, Applications > Sterling Commerce > Yantra Integration
Description	The Yantra adapters represent the functionality available from a specific Yantra system. The adapters are built by connecting to the Yantra system and querying it for a list of available API groups and creating default adapter configurations comprised of those functional groups. The list is then made available to the GPM. Gentran Integration Suite does not know what Yantra functionality is available until the connection is made. All adapters take the same parameters during configuration. Any differences in adapter requirements are contained in the documents sent to Yantra using the given adapter. It is therefore the responsibility of the calling process to deliver the appropriate document to the adapter.
Business usage	Ability to exercise Yantra functionality from within Gentran Integration Suite.
Usage example	A user is configuring a business process graphically. The user selects an adapter from the list of available Yantra adapters, and places it into the business process. After configuration, the adapter is ready to use.
Preconfigured?	No. All adapter configurations are created dynamically at Gentran Integration Suite startup. Default adapter configurations are created dynamically based on API groups. You can create other configurations of any of the adapter types, but do not need to do so. You can use the dynamically created configurations for any needs.
Requires third party files?	Use the install3rdParty script to install yfcb.jar and yfscclient.jar (for your Yantra version). These files are located in Yantra7x/lib. If the Yantra database is different from the Gentran Integration Suite database, you may also need to load the appropriate database jar using the install3rdParty script (for example, instance classes12.zip for Oracle).
Platform availability	<ul style="list-style-type: none">◆ Microsoft Windows◆ Sun Solaris◆ HP-UX◆ IBM-AIX◆ RedHat AS
Related services	None
Application requirements	The requirements are dependent on the API used. See your Yantra documentation for information about the documents passed to each API type.

Initiates business processes?	No
Invocation	The adapters can only be used as part of a business process. The adapters can be located at any point in a business process, but must have the proper business process document to work with the specified API call.
Business process context considerations	These adapters do not initiate a business process. A primary document must always exist for the step that invokes one of the adapters and that is appropriate for the Yantra API call. Any output document received as a result of the API call becomes the new primary document. If any errors occur, the Yantra error code (YFSEException.getErrorCode) is put into ProcessData/Yantra/ErrorCode. The error XML document (YFSEException.getErrorMessage) is put into ProcessData/Yantra/ErrorDetail. The error description (YFSEException.getErrorDescription) is put into the workflow status report.
Returned status values	Success, Error
Restrictions	None
Persistence level	System default
Testing considerations	To test an adapter configuration, create and run a business process that includes it.

How the Yantra Adapters Work

The Yantra adapters are dynamically built each time Gentran Integration Suite is started. Yantra functionality is made available in Gentran Integration Suite through Yantra API groups.

When Gentran Integration Suite starts, a Yantra Startup process runs that creates an adapter for each API group. These default adapter configurations are available in the GPM for use in business processes.

All adapters take the same parameters during configuration. Any differences in adapter requirements are contained in the documents sent to Yantra using the given adapter. The calling process must deliver the appropriate document to the adapter.

Note: See your Yantra documentation for API-specific requirements.

Example

The Yantra system that you are connecting to includes API groups for Inventory Management, Warehouse Layout, Move Requests, Price, and Receipt. When you start your Gentran Integration Suite server, it queries the Yantra database and creates an adapter for each API group it finds. It would create an Inventory Management adapter, Warehouse Layout adapter, Move Requests adapter, Price adapter, and Receipt adapter. These would all be visible and available for use in the GPM.

Implementing the Yantra Adapters

To implement the Yantra adapters, complete the following tasks:

1. Install the third party .jar files. See *Installing the Third Party .jar Files*.
2. Edit the sandbox.cfg file. See *Installing the Third Party .jar Files*.

3. In the GPM, select and configure one or more of the dynamic adapters as part of a business process. See *Configuring the Yantra Adapters*.

Installing the Third Party .jar Files

1. Locate the two Yantra .jar files needed, yfsclient.jar and yfcbe.jar. Ensure that they are the files for your Yantra version. These files are located in Yantra7x/lib.

Note: If the Yantra database is different from the Gentran Integration Suite database, you may also need to load the appropriate database jar using the install3rdParty script (for example, instance classes12.zip for Oracle).

2. Shut down Gentran Integration Suite.
3. From the bin directory where Gentran Integration Suite is installed, install the .jar files by running the install3rdParty script included with Gentran Integration Suite. Use the Yantra server version number and the path and name of the directory where the files are located in the command, using the following examples as a guide:

- ◆ On a Unix system, the commands would look similar to the following:

```
install3rdParty.sh yantra 7x -j directory/yfsclient.jar
```

```
install3rdParty.sh yantra 7x -j directory/yfcbe.jar
```

- ◆ On a Windows system, the commands would look similar to the following:

```
install3rdParty.cmd yantra 7x -j directory/yfsclient.jar
```

```
install3rdParty.cmd yantra 7x -j directory/yfcbe.jar
```

4. After installing the Yantra .jar files, you must add four lines to the sandbox.cfg file, located in the properties subfolder under your Gentran Integration Suite installation folder.

Add the following entries to sandbox.cfg:

- ◆ YANTRA_DB_CLASS=*oracle.jdbc.driver.OracleDriver*
- ◆ YANTRA_DB_URL=*jdbc:oracle:thin:@<host>:<port>:<sid>*
- ◆ YANTRA_DB_USER=*username*
- ◆ YANTRA_DB_PASS=*userpassword*

5. Restart Gentran Integration Suite.

Configuring the Yantra Adapters

In the GPM, use the dynamically created Yantra adapters in your business processes. Each adapter uses the parameters defined in the following table. If one of the adapters requires additional information, you must supply it in the document passed to the adapter in the business process.

Use the field definitions in the following table to set up a Yantra adapter configuration in the GPM. When editing BPML directly (not editing in the GPM), use the names in parentheses to refer to the parameters.

Parameter (BPML value)	Description
Config	Select the name of the adapter configuration from the list.
ApiName (apiName)	Name of the Yantra API to call. Required.
ProgId (progId)	Program identifier used for all transactions in the context of the YFSEnvironment. Required.
Protocol (protocol)	The protocol to use when communicating with the Yantra system. Optional. Valid values are: <ul style="list-style-type: none"> ◆ EJB ◆ HTTP ◆ HTTPS ◆ LOCAL (Yantra is making direct calls to Gentran Integration Suite and no protocol is necessary)
URL (url)	URL of the Yantra system if using HTTP or HTTP/S protocol. Optional.
Userld (userld)	User identifier used for all transaction in the context of the YFSEnvironment. Required.

Parameters Passed From Adapter to Business Process

The following table contains the parameters passed from the Yantra adapter to the business process:

Parameter	Description
Yantra/ErrorCode	The value returned by YFSEnvironment.getErrorCode() if an exception occurs. Only set if YFSEnvironment is thrown.
Yantra/ErrorDetail	The value returned by YFSEnvironment.getErrorDescription() if an exception occurs. Only set if YFSEnvironment is thrown.

Parameters Passed From Business Process to Adapter

The parameters passed from the business process to a Yantra adapter are the same as those listed in *Configuring the Yantra Adapters*.

Business Process Example

The following example illustrates how the Yantra adapter can be used in a business process. In the example, the business process is calling the Yantra Inventory cancelReservation API:

```
<process name = "YantraTest">
  <operation name="Get Inventory Supply">
    <participant name="Yantra_Inventory"/>
    <output message="toService">
      <assign to="." from="*" />
      <assign to="apiName">getInventorySupply</assign>
      <assign to="progId">someProgId</assign>
      <assign to="userId">someUserId</assign>
      <assign to="protocol">HTTP</assign>
      <assign
to="url">http://<host>:<port>/yantra/interop/InteropHttpServlet</assign>
    </output>
    <input message="fromService">
      <assign to="." from="*" />
    </input>
  </operation>
</process>
```

Messages Generated By Yantra Startup Class

The following messages are in noapp.log if the yfscclient.jar is not loaded:

```
Found startup.class1=com.sterlingcommerce.woodstock.services.yantra.YantraStartup
Started com.sterlingcommerce.woodstock.services.yantra.YantraStartup
YantraStartup - Checking for existing service definition file.
YantraStartup - No existing service definition file found.
YantraStartup - Reflecting Yantra APIs.
YantraStartup - Zero API's reflected.
YantraStartup - Finished.
```

This message will always be written to the noapp.log even if you are not using Yantra. Every time Gentran Integration Suite starts, it will check for Yantra.

Suppressing the Messages

If you stop using Yantra with Gentran Integration Suite, or for whatever reason do not want the messages to appear in the log, comment out the following line in noapp.properties by placing a pound sign (#) at the beginning of the line as show below and that class will no longer be called at startup:

```
# startup.class1=com.sterlingcommerce.woodstock.services.yantra.YantraStartup
```

Example Messages

The following are example messages that might be found in noapp.log if yfscclient.jar is loaded and YANTRA_DB_xxx entries are in sandbox.cfg, which would enable Gentran Integration Suite to query the Yantra database when starting up:

```
Found startup.class1=com.sterlingcommerce.woodstock.services.yantra.YantraStartup
Started com.sterlingcommerce.woodstock.services.yantra.YantraStartup
YantraStartup - Checking for existing service definition file.
```

YantraStartup - No existing service definition file found.
YantraStartup - Reflecting Yantra APIss.
YantraStartup - Loading API groups from the database.
YantraStartup - Loading service definition.
YantraStartup - Adding main palette.
DynamicService processing: ../lang/en/Palette_en.properties
YantraStartup - Adding palette item: YantraApi1
YantraStartup - Loading service instance: Yantra Inventory
YantraStartup - Adding palette item: YantraApi2
YantraStartup - Loading service instance: Yantra Order
YantraStartup - Adding palette item: YantraApi3
YantraStartup - Loading service instance: Yantra Payment and Pricing
YantraStartup - Adding palette item: YantraApi4
YantraStartup - Loading service instance: Yantra Receipts
YantraStartup - Adding palette item: YantraApi5
YantraStartup - Loading service instance: Yantra Value Added Services
YantraStartup - Adding palette item: YantraApi6
YantraStartup - Loading service instance: Yantra Product Management
YantraStartup - Adding palette item: YantraApi7
YantraStartup - Loading service instance: Yantra Platform
YantraStartup - Adding palette item: YantraApi8
YantraStartup - Loading service instance: Yantra Delivery Management
YantraStartup - Adding palette item: YantraApi9
YantraStartup - Loading service instance: Yantra Task Management
YantraStartup - Adding palette item: YantraApi10
YantraStartup - Loading service instance: Yantra WMS Inventory
YantraStartup - Adding palette item: YantraApi11
YantraStartup - Loading service instance: Yantra Count
YantraStartup - Adding palette item: YantraApi12
YantraStartup - Loading service instance: Yantra Putaway
YantraStartup - Adding palette item: YantraApi13
YantraStartup - Loading service instance: Yantra Outbound Planning
YantraStartup - Adding palette item: YantraApi14
YantraStartup - Loading service instance: Yantra Warehouse Layout
YantraStartup - Adding palette item: YantraApi15
YantraStartup - Loading service instance: Yantra Picking
YantraStartup - Adding palette item: YantraApi16
YantraStartup - Loading service instance: Yantra Shipping
YantraStartup - Adding palette item: YantraApi17
YantraStartup - Loading service instance: Yantra UnGrouped
YantraStartup - Finished.

Yantra Adapters (Build 4312 or higher)

The Yantra adapters enable you to use Yantra functionality from within Gentran Integration Suite. The following table provides an overview of the Yantra adapters:

Note: You can continue to view and use Yantra adapters configured using Gentran Integration Suite Build 4311 and earlier.

System Name	None.
Graphical Process Modeler (GPM) categories)	All Services.
Description	The Yantra adapters represent the functionality available from a specific Yantra system. You must specify the name of the XAPI you want to connect to you using the Yantra adapter. Gentran Integration Suite does not know what Yantra functionality is available until the connection is made. The adapter requirements are contained in the documents sent to Yantra using the given adapter. It is the responsibility of the calling process to deliver the appropriate document to the adapter.
Business usage	Ability to exercise Yantra functionality from within Gentran Integration Suite.
Usage example	A user is configuring a business process graphically. The user specifies the adapter instance created while configuring a new service and places it into the business process. After configuration, the adapter is ready to use.
Preconfigured?	No. Configure a new service for the Yantra adapter. <ol style="list-style-type: none">1. From the Administration menu, select Deployment > Services > Configuration.2. Select Create > New Service.3. For Service Type, choose YantraAPI, then click Next.
Platform availability	<ul style="list-style-type: none">◆ Microsoft Windows◆ Sun Solaris◆ HP-UX◆ IBM-AIX◆ RedHat AS
Related services	None
Application requirements	The requirements are dependent on the API used. See your Yantra documentation for information about the documents passed to each API type.
Initiates business processes?	No
Invocation	The adapters can only be used as part of a business process. The adapters can be located at any point in a business process, but must have the proper business process document to work with the specified API call.

Business process context considerations	These adapters do not initiate a business process. A primary document must always exist for the step that invokes one of the adapters and that is appropriate for the Yantra API call. Any output document received as a result of the API call becomes the new primary document. If any errors occur, the Yantra error code (YFSEException.getErrorCode) is put into ProcessData/Yantra/ErrorCode. The error XML document (YFSEException.getErrorMessage) is put into ProcessData/Yantra/ErrorDetail. The error description (YFSEException.getErrorDescription) is put into the workflow status report.
Returned status values	Success, Error
Restrictions	None
Persistence level	System default
Testing considerations	To test an adapter configuration, create and run a business process that includes it.

How the Yantra Adapters Work

You must specify the name of the XAPI you want to connect to using the Yantra adapter. Application does not know what Yantra functionality is available until the connection is made. The adapter requirements are contained in the documents sent to Yantra using the given adapter. It is the responsibility of the calling process to deliver the appropriate document to the adapter.

Note: See your Yantra documentation for API-specific requirements.

Configuring the Yantra Adapters

Use the field definitions in the following table to set up a Yantra adapter configuration in the GPM. When editing BPML directly (not editing in the GPM), use the names in parentheses to refer to the parameters. If the adapter requires additional information, you must supply it in the document passed to the adapter in the business process.

Parameter (BPML value)	Description
Config	Specify the name of the adapter configuration.
ApiName (apiName)	Name of the Yantra API to call. Required.
ProgId (progId)	Program identifier used for all transactions in the context of the YFSEnvironment. Required.

Parameter (BPML value)	Description
Protocol (protocol)	The protocol to use when communicating with the Yantra system. Optional. Valid values are: <ul style="list-style-type: none"> ◆ EJB ◆ HTTP ◆ HTTPS ◆ LOCAL (Yantra is making direct calls to Gentran Integration Suite and no protocol is necessary)
URL (url)	URL of the Yantra system if using HTTP or HTTP/S protocol. Optional.
Userld (userId)	User identifier used for all transaction in the context of the YFSEnvironment. Required.

Parameters Passed From Adapter to Business Process

The following table contains the parameters passed from the Yantra adapter to the business process:

Parameter	Description
Yantra/ErrorCode	The value returned by YFSEnvironment.getErrorCode() if an exception occurs. Only set if YFSEnvironment is thrown.
Yantra/ErrorMessage	The value returned by YFSEnvironment.getErrorMessage() if an exception occurs. Only set if YFSEnvironment is thrown.

Parameters Passed From Business Process to Adapter

The parameters passed from the business process to a Yantra adapter are the same as those listed in *Configuring the Yantra Adapters*.

Business Process Example 1

The following example illustrates how the Yantra adapter can be used in a business process. In the example, the business process is calling the Yantra Inventory cancelReservation API. This example is applicable if you have integrated with the Yantra 5.x or 7.x version or Sterling Multi-Channel Selling and Fulfillment (MCF) 8.0 or 8.2:

```
<process name = "YantraTest">
  <operation name="Get Inventory Supply">
    <participant name="<instancename>"/>
```

where <instancename> is the name of the instance you created in GIS.

```
    <output message="toService">
      <assign to="." from="*" />
      <assign to="apiName">getInventorySupply</assign>
      <assign to="progId">someProgId</assign>
      <assign to="userId">someUserId</assign>
      <assign to="protocol">HTTP</assign>
      <assign
to="url">http://<host>:<port>/yantra/interop/InteropHttpServlet</assign>
    </output>
    <input message="fromService">
      <assign to="." from="*" />
    </input>
  </operation>
</process>
```

Business Process Example 2

The following example illustrates how the Yantra adapter can be used in a business process. In the example, the business process is calling the Yantra Inventory cancelReservation API. This example is applicable if your have integrated with Sterling MCF 8.5:

```
<process name = "YantraTest">
  <operation name="Get Inventory Supply">
    <participant name="<instancename>"/>
```

where <instancename> is the name of the instance you created in GIS.

```
    <output message="toService">
      <assign to="." from="*" />
      <assign to="apiName">getInventorySupply</assign>
      <assign to="progId">someProgId</assign>
      <assign to="userId">someUserId</assign>
      <assign to="protocol">HTTP</assign>
      <assign to="url">http://<host>:<port>/smcfs/interop/InteropHttpServlet</assign>
    </output>
    <input message="fromService">
      <assign to="." from="*" />
    </input>
  </operation>
</process>
```

Zengin TCP/IP Adapter

Zengin is a standard Japanese communication protocol. The Zengin TCP/IP adapter enables Application to communicate with Zengin servers. The Zengin TCP/IP adapter is comprised of two components:

Zengin Sender and Requester adapter

Zengin Receiver and Responder adapter

During the configuration process, the Zengin TCP/IP adapter requires the use of the Zengin Configuration Import service and a specially-configured instance of the File System adapter.

You can also control certain behaviors by configuring properties in the `zengin.properties` file.

The following table provides an overview of the Zengin TCP/IP adapter:

System name	ZENGIN_SENDER_REQUESTER, ZENGIN_RECEIVER_RESPONDER
Graphical Process Modeler (GPM) category	None
Description	<p>Transfers files to and from remote systems using TCP/IP. The Zengin TCP/IP adapter is configured in two parts, Zengin Sender and Requester and Zengin Receiver and Responder, and each functions differently within a business process.</p> <p>You can use the Zengin Sender and Requester adapter to send data or requests for data to a remote Zengin server. Use this adapter in a business process.</p> <p>The Zengin Receiver and Responder adapter receives a file or request for data from a remote Zengin server, and then runs a predefined business process to respond to the request accordingly. It acts as a facilitator of incoming requests.</p>
Preconfigured?	No
Requires third party files?	No
Platform availability	<ul style="list-style-type: none"> ◆ Microsoft Windows ◆ Sun Solaris ◆ HP-UX ◆ IBM-AIX ◆ SuSE ◆ RedHat EL
Related services	Zengin Configuration Import service
Application requirements	<p>The Zengin TCP/IP adapter supports Zengin TCP/IP version 2.</p> <p>The application must provide at least one document in process data if a "send" action is specified.</p>
Initiates business processes?	The Zengin Receiver and Responder adapter initiates a business process when a file is received, or when a request to send a file is received.
Invocation	Runs as part of a business process.

Restrictions	Each instance of the Zengin Receiver and Responder adapter must use a port not used by any other application on the host machine.
--------------	---

How the Zengin TCP/IP Adapter Works

The Zengin TCP/IP adapter uses the Zengin Configuration Import service to populate Zengin TCP/IP adapter configurations into a database. The Zengin Configuration Import service enables the Zengin adapter to function, but is not part of the Zengin TCP/IP adapter. It is created independently. For more information, see *Implementing the Zengin Configuration Import Service Configuration* on page 1754.

When the Application server is running, the Zengin Receiver and Responder adapter listens on a port specified in the adapter configuration. When an incoming request is received from a remote Zengin server, the Zengin Receiver and Responder adapter creates a new instance of the Zengin Receiver and Responder adapter to respond to the request and process it accordingly.

The following sections explain how the Zengin Sender and Requester adapter and the Zengin Receiver and Responder adapter function.

Zengin Sender and Requester Adapter, Sending a File

The following steps summarize how the Zengin Sender and Requester adapter works in a business process within Application when sending a file to a remote Zengin server:

1. The Zengin Sender and Requester adapter retrieves a file from the business process primary document and sends it to a remote Zengin server.
2. After sending files to the remote server, the Zengin Sender and Requester adapter updates the business process with the status of each file it attempted to send. The status of each file is nested within a ZenginResult element, where each file alias appears as a separate element with a status value:
 - ◆ success – The file was successfully sent.
 - ◆ fail – The file transmission failed.
 - ◆ not sent – The Zengin TCP/IP adapter did not send the file because of an error.
 - ◆ unknown – The Zengin TCP/IP adapter is unable to determine the status of the file.

View the status report in the Application Business Process Detail window. The following example shows the structure of the status report and the four possible status values:

```
<ZenginResult>
  <file_alias_1>success</file_alias_1>
  <file_alias_2>fail</file_alias_2>
  <file_alias_3>not sent</file_alias_3>
  <file_alias_4>unknown</file_alias_4>
</ZenginResult>
```

Zengin Sender and Requester Adapter, Requesting a File

When the Zengin Sender and Requester adapter requests a file from a remote Zengin server the following steps occur:

1. The Zengin Sender and Requester adapter sends a request for a file to a remote Zengin server.

2. The remote Zengin server replies by sending the requested file.
3. The Zengin Sender and Requester adapter then puts the received file in the business process primary document.

Zengin Receiver and Responder Adapter, Receiving a File

When the Zengin Receiver and Responder adapter receives a file, the following steps occur:

1. The Zengin Receiver and Responder adapter listens on a socket for an incoming message.
2. The Zengin Receiver and Responder adapter receives a file from a remote Zengin server.
3. The adapter reads the settings from the `zengin.properties` file.
4. If the file received has variable-length records, the document does not retain the Zengin protocol record length fields. You must specify whether the file is text or binary. If the file is text, you must also specify whether or not to append a CRLF at the end of each record. For more information, see *Configuring the zengin.properties File* on page 1763.
5. The Zengin Receiver and Responder adapter starts a predefined business process based on the file it receives, and puts the file into the primary document of the business process.
6. The adapter sets two values in the business process, `ZenginFilename` and `ZenginFileAlias`. These correspond to the `filename` and `file_alias` elements in the Zengin configuration XML file.

Zengin Receiver and Responder Adapter, Responding to a File Request

When the Zengin Receiver and Responder adapter responds to a request for a file, the following steps occur:

1. The Zengin Receiver and Responder adapter receives a request from a remote Zengin server to send a file.
2. The Zengin Receiver and Responder adapter starts a predefined business process to retrieve the requested file.
3. The adapter sets two values in the business process that is running, `ZenginFilename` and `ZenginFileAlias`. These correspond to the `filename` and `file_alias` elements, respectively, in the Zengin XML configuration file.
4. If the request is for a text file containing variable length records, then the boundary between records in the file is determined based on the CRLF options. For more information, see *Configuring the zengin.properties File* on page 1763.

Note: Previous versions of the Zengin Receiver and Responder adapter required each record in the file to be preceded by a two-byte Zengin protocol record length in binary format. If the file document to be retrieved by the business process is an older version that contains the Zengin protocol record length fields, the adapter still supports passing the `ZenginVariableRecLenPreInserted` parameter and setting it to “true” in the business process.

5. The business process places the file into its primary document.
6. The Zengin Receiver and Responder adapter extracts the file and sends it to the remote Zengin server.

Implementing the Zengin TCP/IP Adapter

To implement the Zengin TCP/IP adapter, complete the following tasks:

1. Activate your license for the Zengin TCP/IP adapter. For information, see the *An Overview of Implementing Services* topic.
2. Create a Zengin Configuration Import service configuration. For information, see the *Implementing the Zengin Configuration Import Service Configuration* on page 1754.
3. Configure the Zengin XML configuration file. For information, see *Configuring the Zengin XML Configuration File* on page 1757.
4. Create a Zengin Sender and Requester adapter and a Zengin Receiver and Responder adapter configuration. For information, see *Configuring the Zengin TCP/IP Adapter* on page 1761.
5. Use the Zengin Sender and Requester adapter in a business process to send or request a file.

Implementing the Zengin Configuration Import Service Configuration

The Zengin Configuration Import service imports the Zengin XML configuration file into the data source. Creating a Zengin XML configuration is a prerequisite step that is required to use the Zengin TCP/IP adapter.

Note: See *Using the zenginConfigHelper Utility* on page 1761 for information about a command-line utility that can be used instead of the Zengin Configuration Import Service.

To implement the Zengin Configuration Import service:

1. Create the Zengin Configuration Import service configuration. For information, see the *Creating a Service Configuration* topic.
2. Configure the Zengin Configuration Import service. For information, see *Configuring the Zengin Configuration Import Service* on page 1754.
3. Create a File System adapter configuration to read the Zengin XML configuration file. For information, see *Configuring the File System Adapter* on page 1755.
4. Create the Zengin TCP/IP adapter configuration. For information, see the *Creating a Service Configuration* topic.
5. Configure the Zengin TCP/IP adapter. For information, see *Configuring the Zengin TCP/IP Adapter* on page 1761.
6. Create a business process to start the Zengin Configuration Import service, the File System adapter, and the Zengin TCP/IP adapter configurations.

Configuring the Zengin Configuration Import Service

To configure the Zengin Configuration Import service, you must specify field settings in Application.

Application Configuration

The following table describes the fields used to configure the Zengin Configuration Import service in Application:

Field	Description
Name	Unique, meaningful name for the Zengin Configuration Import service configuration.
Description	Meaningful description for the service configuration, for reference purposes.
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see the <i>Using Service Groups</i> topic.</p>

GPM Configuration

There is no configuration required in the GPM for the Zengin Configuration Import service.

Configuring the File System Adapter

To configure the File System adapter instance for use with the Zengin Configuration Import service, you must specify field settings in Application and in the GPM.

Application Configuration

The following table describes the fields used to configure the required File System adapter instance in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique, meaningful name for the File System adapter configuration.
Description	Meaningful description for the adapter configuration, for reference purposes.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see the <i>Using Service Groups</i> topic.</p>
Collection Folder (collectionFolder)	<p>Any folder or subfolder (on the same computer where Application is installed) where files are collected.</p> <p>Do not specify a folder containing programs or any files you do not want to lose. The File System adapter does not copy the files it collects for processing.</p>
Filename Filter (filter)	Leave blank
Collect files from subfolders within and including the collection folder? (useSubFolders)	Select No
Use the absolute file path name for the document name? (keepPath)	Select No
Start a business process once files are collected? (bootstrap)	Select No
Extraction Folder (ExtractionFolder)	Folder where you save the file to write. This is the folder where you placed the Zengin configuration file.
Unobscure File Contents (unobscure)	Select No
Filenaming convention (assignFilename)	Select Assign a specific name
Filename (assignedFilename)	Zengin configuration filename. Required.

GPM Configuration

The business process runs, respectively, the File System adapter configuration and the Zengin Configuration Import service configuration. The first operation of the business process runs the File System adapter to read the Zengin XML configuration file. Specify the following field settings for the File System adapter in the GPM:

Note: In cases where the BPML field name differs from the GPM field name, the BPML field name is shown in parentheses.

Field	Description
Config (participant name)	Name of the adapter configuration. Required.
Action	Action that the File System adapter is to perform. Select Collection (FS_COLLECT). Required.
DeleteAfterCollect	Select No, so the business process does not delete the file after reading it. Required.

The second operation of the business process runs the Zengin Configuration Import service to extract the configuration from the file into the data source or database. There is no need to specify any specific entry. The following business process is an example of how this operation should look:

```
<process name="zengin_configuration_import">
<sequence name="Main sequence">
  <operation name="File System Adapter">
    <participant name="my_file_sys_adapter_instance"/>
    <output message="FileSystemInputMessage">
      <assign to="Action">FS_COLLECT</assign>
      <assign to="assignedFilename" from=" `my_zengin_config_xml' "></assign>
      <assign to="collectionFolder">my_folder</assign>
      <assign to="deleteAfterCollect">>false</assign>
      <assign to="filter" from='my_zengin_config_xml' "></assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <operation name="Zengin Configuration Import Service">
    <participant name="my_zengin_config_import_instance"/>
    <output message="output">
      <assign to="." from="*"></assign>
    </output>
    <input message="input">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</sequence>
</process>
```

An example of this business process (ZenginConfigImport.bpml) is available in the *installroot/samples/Zengin* folder.

Configuring the Zengin XML Configuration File

The XML schema for the Zengin configuration file is defined in the file, *zengin-config.xsd*, which is included with Application. Find this file in the Application DTD/Schema repository.

The following table describes the format of the elements in the Zengin XML configuration file. You must preconfigure these elements for the Zengin TCP/IP adapter to work properly. Several of the element values are represented as hexadecimal (base 16) numbers because both ROS3 Zengin servers and CJS (Chori Joho Systems) Zengin servers use hexadecimal notation for these elements.

XML Element	Description
zengin_id	When nested within the server element, indicates the local server. When nested within the client element indicates the remote servers. This element can appear only once within the server element, and can appear multiple times within the client element.
server	Definition of a single local server and all of its associated remote servers. The Zengin TCP/IP adapter supports only one server element appearing within the Zengin XML configuration file.
client	All of the remote Zengin server definitions.
password	Use the same password for both directions of communication between Application and a remote Zengin server. Must be 12 hexadecimal digits; use leading 0s if necessary.
alias	Identifies the local and remote Zengin servers. The alias for the local server must be <i>localhost</i> . The alias for a remote server may be any value.
center_code	Entity sending or receiving files. There is one code for the remote Zengin server and the local Application instance. Must be 10 hexadecimal digits.
cpu_node_code	Sequence number for each CPU/Node actually connected as if to a line and used for data exchange. Must be four hexadecimal digits; use leading 0s if necessary.
file_alias	Identifies the file being sent or requested. Human readable alias for a filename. Each file alias associated with a particular client must be unique; do not use the same file_alias multiple times within the client element.
filename	Name of the file being sent or requested. Represents an industry code and file type. Each filename associated with a particular client must be unique; do not use the same filename multiple times within the client element. Valid value is 24 hexadecimal digits
receive_BP_name	Name of the business process to start when a file is received from a remote Zengin server.
request_BP_name	Name of the business process to start when a file is requested by a remote Zengin server.
record_length	Length of the record in bytes. When sending a file or responding to a request for a file: <ul style="list-style-type: none"> ◆ record_length is required. ◆ If fixed_record_length is set to true, then $1 < \text{record_length} \leq 2043$, and record_length must be a factor of file size. ◆ If fixed_record_length is set to false, then $0 < \text{record_length} \leq 2041$. When requesting a file: <ul style="list-style-type: none"> ◆ If fixed_record_length is set to true, then record_length is required. ◆ If fixed_record_length is set to false, then record_length is optional and can be set to 0. When receiving a file, the record_length value is not used.
fixed_record_length	Whether each record in the file has the same length. Valid values are true and false. If false, each record may have a different length.

XML Element	Description
file_access_key	Used for access control as a part of data protection. Must be 12 hexadecimal digits; use leading 0s if necessary.
use_compression	Valid values are true and false.

The following sample Zengin XML configuration file defines the required single instance of the local server (<alias>localhost</alias>), a single remote Zengin server (<alias>Ros3</alias>), and two files associated with the ROS3 server:

```
<?xml version="1.0" encoding="UTF-8"?>
<zengin xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="zengin-config.xsd">
  <server>
    <zengin_id>
      <alias>localhost</alias> <!--the local server(Application)-->
      <center_code>1212121212</center_code>
      <cpu_node_code>1A1B</cpu_node_code>
    </zengin_id>
    <client>
      <zengin_id>
        <alias>Ros3</alias>
        <center_code>123456789A</center_code>
        <cpu_node_code>F1F1</cpu_node_code>
      </zengin_id>
      <password>FFFFFFFFFFFF</password>
      <file_definition>
        <file_alias>send_variable_uncompr</file_alias>
        <filename>FOFOFOFOFOFOFOFOFOFOFOFOFOFO</filename>
        <request_bp_name>bpreq</request_bp_name>
        <receive_bp_name>bprecv</receive_bp_name>
        <fixed_record_length>>false</fixed_record_length>
        <record_length>256</record_length>
        <use_compression>>false</use_compression>
        <file_access_key>AA00000000AA</file_access_key>
      </file_definition>
      <file_definition>
        <file_alias>send_fixed_uncompr</file_alias>
        <filename>0000000000000000000000000001</filename>
        <request_bp_name>bpreq</request_bp_name>
        <receive_bp_name>bprecv</receive_bp_name>
        <fixed_record_length>>true</fixed_record_length>
        <record_length>256</record_length>
        <use_compression>>false</use_compression>
        <file_access_key>FE00120000EF</file_access_key>
      </file_definition>
    </client>
  </server>
</zengin>
```

Securing the Zengin XML Configuration File

Information is normally stored in the Zengin XML configuration file in clear text. This includes the client alias passwords and file access keys. This may or may not be acceptable. In cases where clear text is not acceptable, you should secure that data.

Application includes a utility called `zenginSecureConfig` that obscures (encrypts) the client alias passwords and file access keys contained in the Zengin XML configuration file.

To run the `zenginSecureConfig` utility, use one of the following commands from the command line:

```
install_dir/bin/zenginSecureConfig.sh configFileName [encoding] (UNIX)
```

```
install_dir/bin/zenginSecureConfig.cmd configFileName [encoding] (Windows)
```

Specify the path to the Application installation directory for `install_dir` and the Zengin XML configuration file name for `configFileName`. The [encoding] parameter allows you to enter an optional character encoding. If not entered, the utility uses the default value of UTF-8.

The `zenginSecureConfig` utility reads the specified configuration file and parses it for validity. As the file is parsed, the utility checks for:

1. Client password nodes (`//zengin/client/password`)
 - ◆ If found, you will be prompted to enter a password or to use the existing value (if one is present).
 - ◆ If no password is found for a given client node, you are prompted to enter a password for that client node.
 - ◆ After you enter a password, a `<password_secure>` node tag is written with the obscured (encrypted) value.
2. File access key nodes (`//zengin/client/file_definition/file_access_key`)
 - ◆ If found, you are prompted to enter a file access key or to use the existing value (if one is present).
 - ◆ If no file access key is found for a given file definition, you are prompted to enter a file access key for that file definition node.
 - ◆ After you enter a file access key, a `<file_access_key_secure>` node tag is written with the obscured (encrypted) value.

When the `zenginSecureConfig` utility is finished running, all client passwords and file access keys are secure and may be safely stored in the system.

To reset any of the secure values, perform one of the following steps:

Remove the secure value, leaving an empty node. (For example, `<password_secure/>` or `<password_secure></password_secure>`).

Remove the “_secure” part of the tag and enter a clear text value or leave the value empty. (For example, `<password/>` or `<password>clearTextPsw</password>`).

Delete the entire secure node.

After the secure values have been reset, run the `zenginSecureConfig` utility, if needed, to secure any new values.

Using the zenginConfigHelper Utility

To export a Zengin XML configuration from the data source to a file, use the `zenginConfigHelper` utility provided with Application. This utility also allows you to import a Zengin XML Configuration file to the data source and can be used instead of the Zengin Configuration Import service for this purpose. The `zenginConfigHelper` utility is especially useful for easily importing configuration files created with the `zenginSecureConfig` utility.

To export a Zengin XML Configuration, run the `zenginConfigHelper` utility using one of the following commands from the command line:

```
install_dir/bin/zenginConfigHelper.sh export configFileName (UNIX)
```

```
install_dir/bin/zenginConfigHelper.cmd export configFileName (Windows)
```

To import a Zengin XML Configuration, run the `zenginConfigHelper` utility using one of the following commands from the command line:

```
install_dir/bin/zenginConfigHelper.sh import configFileName (UNIX)
```

```
install_dir/bin/zenginConfigHelper.cmd import configFileName (Windows)
```

Specify the path to the Application installation directory for `install_dir` and the Zengin XML configuration file name for `configFileName`.

Configuring the Zengin TCP/IP Adapter

To configure the Zengin TCP/IP adapter, you must configure both the Zengin Sender and Requester adapter and the Zengin Receiver and Responder adapter by specifying field settings in Application and in the Graphical Process Modeler. For general information about service and adapter configurations, see the *Creating a Service Configuration* topic.

Configuring the Zengin Sender and Requester Adapter

Application Configuration

You can create as many configurations of the Zengin Sender and Requester adapter as necessary. If you leave the optional parameters in the following table blank, you can use a single adapter instance for multiple remote Zengin servers.

The following table describes the fields used to configure the Zengin Sender and Requester adapter in Application:

Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.

Field	Description
Select a Group	<p>Select one of the options:</p> <ul style="list-style-type: none"> ◆ None – You do not want to include this configuration in a group at this time. ◆ Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. ◆ Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. <p>Note: For more information about groups, see the <i>Using Service Groups</i> topic.</p>
Zengin Server Alias (ZenginRemoteServer Alias)	Identifies the Zengin destination server with which the Zengin TCP/IP adapter communicates. The alias identifies such values as Hostname, Port, Center Code, and CPU/Node code. Must match the value of the <code>//client/zengin_id/alias</code> element in the Zengin XML configuration file. Optional, however, if left blank, you must specify the Zengin server alias in the GPM or at runtime in process data.
Hostname or IP (ZenginHostname)	Remote Zengin server IP address or DNS name. Must correspond with the server specified in Zengin Server Alias . Optional, however, if left blank, you must specify in the GPM or at runtime in process data.
Port (ZenginPort)	Remote Zengin server port number. Optional, however, if left blank, you must specify in the GPM or at runtime in process data.

GPM Configuration

The following table describes the fields used to configure the Zengin Sender and Requester adapter in the GPM:

Field	Description
Config	Name of the adapter configuration.
ZenginDocuments	<p>When sending multiple files, this XPath expression identifies all documents in process data which will be sent. All XML element names must be equal to the appropriate file alias.</p> <p>For example, if process data contains:</p> <pre><myDocs> <FileAlias1 SCIObjctID="someDbObjctId"/> <FileAlias2 SCIObjctID="anotherDbObjctId"/> </myDocs></pre> <p>and the business process contains the following:</p> <pre><assign to="ZenginDocuments">//myDocs/*</assign></pre> <p>You must configure FileAlias1 and FileAlias2 as valid file aliases (using the Zengin XML configuration file).</p>
ZenginFileAlias	<p>When sending a file, used to look up all other file configuration values, access key, record type, and compression.</p> <p>For example: <code><assign to="ZenginFileAlias">myAlias</assign></code></p> <p>In this case, you must configure myAlias as a file alias using the Zengin XML configuration file.</p>

Field	Description
ZenginHostname	Remote Zengin server IP address or DNS name. Must correspond with the server specified by ZenginRemoteServerAlias . If left blank in the Application configuration, you must specify this parameter either here or at runtime in process data.
Zengin Mode	Mode is either Send file or Request file. The corresponding business process parameter values are send and request. Sender parameter.
ZenginPort	Remote Zengin server port number. If left blank in the Application configuration, you must specify this parameter either here or at runtime in process data.
ZenginRemoteServerAlias	Identifies the Zengin destination server with which the Zengin TCP/IP adapter communicates. The alias identifies such values as Hostname, Port, Center Code, and CPU/Node code. Must match the value of the <code>//client/zengin_id/alias</code> element in the Zengin XML configuration file. If left blank in the Application configuration, you must specify this parameter either here or at runtime in process data.
ZenginVariableRecLenPreInserted	Either true or false. If true, the file being sent has variable length records, and the two-byte record length is pre-inserted in the data before each record. Sender parameter. Note: This field is no longer necessary, but is retained for compatibility with older files preceded with a 2-byte record length.

Configuring the Zengin Receiver and Responder Adapter

Application Configuration

Due to TCP/IP requirements, only one configuration of the Zengin Receiver and Responder is allowed for a particular port. The following table describes the field used to configure the Zengin Receiver and Responder adapter in Application:

Field	Description
Listen Port	Remote Zengin server port number. Required. Valid values: 1025 – 65535 The default value is 5020.

GPM Configuration

There is no configuration required in the GPM for the Zengin Receiver and Responder adapter.

Configuring the zengin.properties File

The `zengin.properties` file in the `properties` directory provides settings that control the retry behavior of the Zengin TCP/IP adapter and how it handles files containing variable-length records.

To configure the `zengin.properties` file, perform the following steps:

1. Locate the zengin.properties.in file in the properties directory where you installed Application.
2. Open the zengin.properties.in file in a text editor.
3. Configure the properties according to the following tables:

Specify the following default settings in the zengin.properties.in file:

Property	Description
default.Timeout	Timeout value, in seconds, if the Zengin TCP/IP adapter was idle during a receive process.
Note: The following properties control the Retry behavior of the Zengin TCP/IP adapter.	
default.AutoRetry	Specifies whether the adapter should make retry attempts if the connection between the adapter and the remote Zengin server is broken during a send or receive process. Valid values: <ul style="list-style-type: none"> ◆ true—Automatically make retry attempts ◆ false—Do not make retry attempts (default)
default.MaxRetry	Maximum number of retry attempts to make. Valid values are any positive integer. This property is only used if <i>AutoRetry</i> is set to true. Default is 3.
default.RetryInterval	Interval, in seconds, between each retry attempt. Valid values are any integer greater than 1. This property is only used if <i>AutoRetry</i> is set to true and <i>MaxRetry</i> is greater than 1. Default is 3 seconds.

You should always specify default settings. If you need different settings for a particular station and file, you can also configure each station and file combination using this syntax:

StationAlias.FileAlias.property=setting

For example, if you have the default.AutoRetry set to true, but you do not want retry attempts made for FILE01 in STATION01 as defined in the Zengin XML configuration file, enter STATION01.FILE01.AutoRetry=false in the zengin.properties.in file. This will override the default setting for that station and file only.

You can also specify settings for each station and file combination to control how the Zengin TCP/IP adapter handles files containing variable-length records.

For each specific Station and File combination, you can specify the settings in the following table:

Note: In all the following properties, *StationAlias* and *FileAlias* must match the zengin XML configuration file.

Property	Description
<i>StationAlias.FileAlias.Timeout</i>	Timeout value, in seconds, if the Zengin TCP/IP adapter was idle during a receive process.
Note: The following properties control the Retry behavior of the Zengin TCP/IP adapter.	

Property	Description
<i>StationAlias.FileAlias.AutoRetry</i>	Specifies whether the Zengin TCP/IP adapter should make retry attempts if the connection between the adapter and the remote Zengin server is broken during a send or receive process. Valid values: <ul style="list-style-type: none"> ◆ true—Automatically make retry attempts ◆ false—Do not make retry attempts
<i>StationAlias.FileAlias.MaxRetry</i>	Maximum number of retry attempts to make. Valid values are any positive integer. This property is only used if AutoRetry is set to true.
<i>StationAlias.FileAlias.RetryInterval</i>	Interval, in seconds, between each retry attempt. Valid values are any integer greater than 1. This property is only used if AutoRetry is set to true and MaxRetry is greater than 1.
	Note: The following properties control the handling of files containing variable-length records.
<i>StationAlias.FileAlias.FileType</i>	Specifies the type of file. Valid values: <ul style="list-style-type: none"> ◆ binary—File content will not be altered before being sent or received ◆ text—File content will be altered according to the AppendCRLF and CRLFHexCode property settings Default is binary.
<i>StationAlias.FileAlias.AppendCRLF</i>	In a text file, specifies whether to append each record with a delimiter. Valid values: <ul style="list-style-type: none"> ◆ true—When received, each record in the file will be appended with the hexadecimal code specified in CRLFHexCode. When being sent, the code will be removed from the end of each record. ◆ false—The records will not be appended. Default is false.
<i>StationAlias.FileAlias.CRLFHexCode</i>	Specifies the delimiter to append to each record in a text file. Valid values: <ul style="list-style-type: none"> ◆ 0A—Appends a line feed (new line) to the end of each record ◆ 0D0A—Appends a line feed and a carriage return to the end of each record Default is 0A. This property is only used if FileType is set to text and AppendCRLF is set to true.

4. Save and close the zengin.properties.in file.
5. Run one of the following utilities to update the zengin.properties file:
 - ◆ `/install_dir` (UNIX)
 - ◆ `install_dir\bin\setupfiles.cmd` (Windows)
6. Stop and restart Application to use the updated settings.

Example Business Processes

Receive Example

This business process is called when a file is received and it saves the file to disk. Note that the Zengin TCP/IP adapter invokes the business process when a file is received, the business process does not invoke the adapter.

```
<process name="zengin_file_sys_write">
  <sequence name="Main Sequence">
    <operation name="File System Adapter">
      <participant name="my_file_sys"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="assignedFilename">received_filename</assign>
        <assign to="extractionFolder">/my_directory</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Respond Example

This business process is called when the Zengin TCP/IP adapter receives and responds to a request to send a file. The business process reads a file from the disk to be sent back to the remote Zengin server. Note that the Zengin TCP/IP adapter invokes the business process when a request is received. The business process does not invoke the adapter.

```
<process name = "zengin_file_sys_read">
  <sequence name="Main Sequence">
    <operation name="File System Adapter">
      <participant name="my_file_sys"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="assignedFilename">respond_filename</assign>
        <assign to="collectionFolder">/my_directory</assign>
        <assign to="deleteAfterCollect">>false</assign>
        <assign to="filter">respond_fixed_uncompr</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Send Example

This business process sends a single file to a remote Zengin server.

```

<process name = "zengin_sender">

  <sequence name="Main Sequence">

    <operation name="send fixed uncompr">
      <participant name="zengin_sender"/>
      <output message="output">
        <assign to="ZenginRemoteServerAlias">Ros3</assign>
        <assign to="ZenginFileAlias">send_fixed_uncompr</assign>
        <assign to="ZenginMode">send</assign>
        <assign to="ZenginHostname">123.45.67.8</assign>
        <assign to="ZenginPort">5020</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="send var uncompr">
      <participant name="zengin_sender"/>
      <output message="output">
        <assign to="ZenginRemoteServerAlias">Ros3</assign>
        <assign to="ZenginFileAlias">send_variable_uncompr</assign>
        <assign to="ZenginMode">send</assign>
        <assign to="ZenginHostname">123.45.67.8</assign>
        <assign to="ZenginPort">5020</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>

```

Send Multiple Files Example

This business process sends several files to a remote Zengin server.

```

<process name = "zengin_send_mult">

  <sequence name="Main Sequence">

    <assign to="myDocs/send_fixed_uncompr" from="PrimaryDocument/@*"
    append="true"></assign>
    <assign to="myDocs/send_variable_uncompr" from="PrimaryDocument/@*"
    append="true"></assign>

    <operation name="call sender">
      <participant name="zengin_sender"/>
      <output message="output">
        <assign to="ZenginRemoteServerAlias">Ros3</assign>
        <assign to="ZenginDocuments">../myDocs/*</assign>
        <assign to="ZenginMode">send</assign>
      </output>
    </operation>
  </sequence>
</process>

```

```

    <assign to="ZenginHostname">123.45.67.8</assign>
    <assign to="ZenginPort">5020</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="input">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</process>

```

Request Files Sample

This business process requests files from a remote Zengin server.

```

<process name = "zengin_requester">
  <sequence name="Main Sequence">
    <operation name="fixed uncompr">
      <participant name="zengin_sender"/>
      <output message="output">
        <assign to="ZenginRemoteServerAlias">Ros3</assign>
        <assign to="ZenginFileAlias">request_fixed_uncompr</assign>
        <assign to="ZenginMode">request</assign>
        <assign to="ZenginHostname">123.45.67.8</assign>
        <assign to="ZenginPort">5020</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="File System Adapter">
      <participant name="file_sys"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="assignedFilename">request_fixed_uncompr</assign>
        <assign to="extractionFolder">/myDirectory</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="variable uncompr">
      <participant name="zengin_sender"/>
      <output message="output">
        <assign to="ZenginRemoteServerAlias">Ros3</assign>
        <assign to="ZenginFileAlias">request_var_uncompr</assign>
        <assign to="ZenginMode">request</assign>
        <assign to="ZenginHostname">123.45.67.8</assign>
        <assign to="ZenginPort">5020</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```



```

</input>
</operation>

<operation name="File System Adapter">
  <participant name="file_sys"/>
  <output message="FileSystemInputMessage">
    <assign to="Action">FS_EXTRACT</assign>
    <assign to="assignedFilename">request_var_uncompr</assign>
    <assign to="extractionFolder">/myDirectory</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="fixed compr">
  <participant name="zengin_sender"/>
  <output message="output">
    <assign to="ZenginRemoteServerAlias">Ros3</assign>
    <assign to="ZenginFileAlias">request_fixed_compr</assign>
    <assign to="ZenginMode">request</assign>
    <assign to="ZenginHostname">123.45.67.8</assign>
    <assign to="ZenginPort">5020</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="input">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="File System Adapter">
  <participant name="file_sys"/>
  <output message="FileSystemInputMessage">
    <assign to="Action">FS_EXTRACT</assign>
    <assign to="assignedFilename">request_fixed_compr</assign>
    <assign to="extractionFolder">/myDirectory</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

<operation name="variable compr">
  <participant name="zengin_sender"/>
  <output message="output">
    <assign to="ZenginRemoteServerAlias">Ros3</assign>
    <assign to="ZenginFileAlias">request_var_compr</assign>
    <assign to="ZenginMode">request</assign>
    <assign to="ZenginHostname">123.45.67.8</assign>
    <assign to="ZenginPort">5020</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="input">
    <assign to="." from="*"></assign>
  </input>
</operation>

```

```
</input>
</operation>

<operation name="File System Adapter">
  <participant name="file_sys"/>
  <output message="FileSystemInputMessage">
    <assign to="Action">FS_EXTRACT</assign>
    <assign to="assignedFilename">request_var_compr</assign>
    <assign to="extractionFolder">/myDirectory</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

</sequence>
</process>
```

ZlibInflate Service

The following table provides an overview of the ZlibInflate service:

System name	ZlibInflate
Graphical Process Modeler (GPM) category	All Services and Applications > ERP
Description	Decodes and decompresses messages received from PeopleSoft.
Business usage	The ZlibInflate service may only be used with the Adapter for PeopleSoft as it decodes and uncompresses response message received from the PeopleSoft system.
Usage example	A configuration of the ZlibInflate service has to be created. No parameters are required during configuration creation or inside the BPML. After creating the configuration, you can use the service in a business process.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Application platforms
Related services	Adapter for PeopleSoft
Application requirements	A configuration of the ZlibInflate service has to be created for use inside a BPML service definition. No configuration or BPML parameters are required.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	No
Returned status values	Basic Status – Success or Failure No advanced statuses
Restrictions	The ZlibInflate service is instantiated at runtime by the business process engine, so a single configuration may be defined for use by multiple business processes at the same time (as the business process engine instantiates a new object for each business process in which they run).
Testing considerations	Test as part of a Adapter for PeopleSoft implementation.

Business Process Example

The following example shows how the ZlibInflate service can run by a business process:

```
<operation name="Inflate incoming PS message">
<participant name="ZlibInflate"/>
  <output message="Xout">
```

```
<assign to="." from="*" />
</output>
<input message="Xin">
<assign to="." from="*" />
</input>
</operation>
```