

Sterling Integrator®

Services and Adapters (M to Z)

Version 5.1



Contents

Mail Mime Service.....	6
Mailbox Add Service.....	14
Mailbox Correlate Document Service.....	19
Mailbox Delete Mailbox Service.....	21
Mailbox Delete Service.....	27
Mailbox Evaluate All Automatic Routing Rules Service.....	31
Mailbox Evaluate Routing Rule Service.....	35
Mailbox Extract Abort Service.....	40
Mailbox Extract Begin Service.....	44
Mailbox Extract Commit Service.....	48
Mailbox List Service.....	52
Mailbox Query Service.....	54
Mailbox Scheduled Delete Service.....	61
Mailbox Update Service.....	65
Mapped Extraction Service.....	68
MergeDocument Service.....	71
MIME Service	74
MQRFH2 Service.....	89
MSMQ Adapter.....	92
MSMQ Send Service.....	99
Network Report Reconciliation Service.....	105
Obscure Data - Reveal Primary Document Service.....	109
Obscure Data - Obscure Primary Document Service.....	112
Obscure Data - Process Data Values Service.....	115
Odette FTP Adapter.....	120
Odette FTP Queue Handler Service.....	134
OdetteFTP Scheduler Service.....	143
Oftp Adapter.....	145
Oracle AQ JMS Queue Adapter.....	152

Oracle AQ JMS Topic Adapter.....	156
Oracle E-Business Suite Adapter Configuration Service.....	160
Oracle E-Business Suite Message Service.....	165
Oracle E-Business Suite Adapter.....	170
Adapter for PeopleSoft.....	180
PGP Package Service.....	198
PGP Unpackage Service.....	212
Adapter for PeopleSoft CRM CIC.....	222
Purge Business Process Linkage Service.....	233
Purge Service.....	235
Release Service	239
Report Service.....	246
Restore Business Process Service.....	248
Remote Method Invocation (RMI) Adapter.....	251
SAP XI Adapter.....	257
SAP Suite Adapter	279
Script Adapter.....	354
SFTP Client Adapter.....	359
SFTP Client Begin Session Service.....	369
SFTP Client CD Service.....	376
SFTP Client DELETE Service.....	379
SFTP Client End Session Service.....	382
SFTP Client GET Service.....	385
SFTP Client LIST Service.....	388
SFTP Client MOVE Service.....	392
SFTP Client PUT Service.....	395
SFTP Client PWD Service.....	399
SFTP Server Adapter.....	402
Sterling Information Broker Adapter.....	413
Sleep Service.....	418
SMTP Send Adapter.....	421
Simple Network Management Protocol (SNMP) Trap Adapter.....	437

SOA Inbound Message Processing Service.....	445
SOA Inbound Security Service.....	451
SOA Outbound Message Processing Service.....	457
SOA Outbound Security Service.....	463
SOAP Inbound Service.....	470
SOAP Outbound Service.....	482
SSH Key Grabber Adapter.....	487
SSLCertGrabber Adapter.....	492
Sterling Control Center Service.....	496
Straight Through Extraction Service.....	500
Text Document Appender Service.....	502
This Service	504
TIBCO Rendezvous Adapter.....	519
Timestamp Service for PeopleSoft.....	525
Timestamp Utility Service.....	527
Timezone Offset Service.....	530
Typing Service.....	532
User Service.....	535
Vitria (Businessware) Adapter.....	538
Wait Notify Service.....	544
Wait Service.....	546
webMethods Adapter.....	548
WebSphere MQ Adapter.....	566
WSDL Service.....	587
WebSphereMQ Suite Async Receiver Adapter.....	590
WebSphereMQ Suite Backout Service.....	602
WebSphereMQ Suite Close Queue Service.....	606
WebSphereMQ Suite Close Session Service.....	610
WebSphereMQ Suite Commit Service.....	614
WebSphereMQ Suite Get Message Service.....	618
WebSphereMQ Suite Open Queue Service.....	627
WebSphere MQ Suite Open Session Service.....	632

WebSphereMQ Suite Put Message Service.....	638
XAPI Service.....	647
XML Digital Signature Service.....	652
XML Encoder Service.....	673
XML Encryption Service.....	676
XML Validation Service.....	680
XSLT Service.....	684
Yantra Adapters.....	691
Zengin TCP/IP Adapter.....	698
ZlibInflate Service.....	717

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. 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. The MIME message is then typically sent to the SMTP adapter to mail it. 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. 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 Sterling Integrator platforms
Related services	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	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	You can use the business process examples later in this section to test a configuration of the service. The problems encountered most frequently are: <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*.
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 Sterling Integrator.
- 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.

Sterling Integrator Configuration

The following table describes the fields used to configure the Mail Mime service in Sterling Integrator:

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. <hr/> 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.

Field	Description
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.
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.
<p>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.</p>	
<p>Note: The following parameters are not displayed in the GPM by default, however, you can manually add these parameters when necessary.</p>	
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 n is the attachment count such as 1, 2, 3...n. 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 Sterling Integrator, 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 n is the attachment count. Valid value is a fully-qualified path to a file on the file system. Optional.

Parameter	Description
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.
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 Sterling Integrator
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 Sterling Integrator Mailbox that the message extract request has successfully completed. • Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Sterling Integrator 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 Sterling Integrator 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	Returned status values: <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	Limitations for users and groups. 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.
Testing considerations	Troubleshooting information for this service can be found in the Mailbox log files.

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*.
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>yyyyMMddThmm</i> format. Note: You must provide a value for this field or one of the following: <ul style="list-style-type: none"> • ExtractableCount • Extractable

Field	Description
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/InboundThe following characters cannot be used in a mailbox name: ? < > : \ / % *.
MessageName	User-defined. The following characters cannot be used in a message name: \ / : * ? " < > % !If you do not provide a value for MessageName, Sterling Integrator 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, Sterling Integrator 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.

Parameter	Description
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 Sterling Integrator platforms
Related services	<p>The Mailbox Correlate Document service works with the other Sterling Integrator 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 Sterling Integrator Mailbox that the message extract request has successfully completed.

	<ul style="list-style-type: none"> • Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Sterling Integrator 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 Sterling Integrator Mailbox. • Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Sterling Integrator 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>Returned 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 Sterling Integrator).
Testing considerations	Troubleshooting information for this service can be found in Sterling Integrator Mailbox log files.

How the Mailbox Correlate Document Service Works

Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator platforms
Related services	This service also works with the other Mailbox services to provide a mailbox capability to Sterling Integrator. The related services are: <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.

	<ul style="list-style-type: none"> • 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 Sterling Integrator 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*.
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 Sterling Integrator.

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. <hr/> <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>
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. 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>
UserId	<p>Do not use.</p> <p>Note: The UserId 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 Sterling Integrator 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 Sterling Integrator 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>
      </output message>
    </operation>
  </sequence>
</process>
```



```

    <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>

```

```
</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 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 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 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	Returned 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 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 message>
    </operation>
  </sequence>
</process>
```

```

    </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 or GPM:

Field	Description
DaysOld	Number of days a message can exist in 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 to specify which mailboxes to query. <hr/> 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

Field	Description
	<p>format if a wildcard is used:x.y where x and y may be alphanumeric or a wildcard. Example:*.po</p> <p>Note: You cannot specify a message name pattern composed solely of a single asterisk. it will fail the adapter.</p>
StartDateTime	Defines the earliest allowable value for this query. Valid values are dates and times in <i>yyyyMMddThmm</i> format.
UserId	<p>User ID associated with the business process.</p> <p>Note: The user ID must have access to the mailbox being queried.</p>

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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator platforms

Related services	<p>The Mailbox Evaluate All Automatic Routing Rules service works with the other Sterling Integrator 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 Sterling Integrator Mailbox that the message extract request has successfully completed. • Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Sterling Integrator 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 Sterling Integrator is required to use this service.
Initiates business processes?	No
Invocation	Not applicable (Internal service)
Business process context considerations	None
Returned status values	<p>Returned status values:</p> <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 Sterling Integrator Mailbox log files.

How the Mailbox Evaluate All Automatic Routing Rules Service Works

Sterling Integrator 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*.
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 Sterling Integrator:

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.
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	Information about scheduling the service configuration to run. Valid values: <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.

Field	Description
	<ul style="list-style-type: none">• Run based on day(s) of the month <p>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>

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 Sterling Integrator 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 Sterling Integrator platforms
Related services	<p>The Mailbox Evaluate Routing Rule service works with the other Sterling Integrator 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 Sterling Integrator Mailbox that the message extract request has successfully completed. • Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Sterling Integrator Mailbox. • Mailbox Query service – Enables querying messages.

	<ul style="list-style-type: none"> • Mailbox Delete service – Enables deletion of messages. • Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Sterling Integrator Mailbox.
Application requirements	Nothing external to Sterling Integrator 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	<p>Returned status values:</p> <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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator:

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.

Field	Description
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.
Mailbox Routing Rule(RuleId)	A valid ID for a routing rule. Required if RuleDescription is not provided.
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. 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

Field	Description
	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.
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>
```

```
    </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 Sterling Integrator platforms
Related services	<p>The Mailbox Extract Abort service works with the other Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator Mailbox.

	<ul style="list-style-type: none"> • Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Sterling Integrator 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>Returned 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 Sterling Integrator 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 Sterling Integrator 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*.
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 Sterling Integrator platforms
Related services	<p>The Mailbox Extract Begin service works with the other Sterling Integrator 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 Sterling Integrator Mailbox that the message extract request has successfully completed. • Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Sterling Integrator 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 Sterling Integrator Mailbox. • Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.

Application requirements	Nothing external to Sterling Integrator 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	The possible status values a service can return are: <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 Sterling Integrator 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 Sterling Integrator 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*.
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.
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 Sterling Integrator platforms
Related services	<p>The Mailbox Extract Commit service works with the other Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator Mailbox. • Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.

Application requirements	Nothing external to Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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*.
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
Messageld	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 Sterling Integrator platforms
Related services	<p>The Mailbox List service works with the other Sterling Integrator 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 Sterling Integrator Mailbox that the message extract request has successfully completed. • Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Sterling Integrator 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 Sterling Integrator Mailbox.

	<ul style="list-style-type: none"> • Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.
Application requirements	Nothing external to Sterling Integrator is required to use this service.
Initiates business processes?	No
Invocation	Not applicable (Internal service)
Business process context considerations	No
Returned status values	<p>The possible status values a service can return are:</p> <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 Sterling Integrator 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 Sterling Integrator mailboxes for messages that meet specified criteria and returns results.
Business usage	Use this service to find messages in Sterling Integrator Mailbox.
Usage example	Search for messages in one or more Sterling Integrator mailboxes.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Sterling Integrator platforms
Related services	<p>The Mailbox Query service works with the other Sterling Integrator 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 Sterling Integrator Mailbox that the message extract request has successfully completed. • Mailbox Extract Abort service – Enables a business process to signal a failed message extraction to Sterling Integrator Mailbox. • Mailbox Delete service – Enables deletion of messages. • Mailbox Evaluate All Automatic Rules service – Provides for the scheduling of automatic routing rules for Sterling Integrator Mailbox. • Mailbox Evaluate Routing Rule service – Enables triggering of mailbox routing rules from a business process.

Application requirements	Nothing external to Sterling Integrator 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	Returned 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 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 Sterling Integrator Mailbox log files.

How the Mailbox Query Service Works

Use the Mailbox Query service to query Sterling Integrator 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>
    </operation>
  </sequence>
</process>
```

```

    </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*.
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:

Field	Description
	? < > / " % *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.
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 Sterling Integrator.

Parameter	Description
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.
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.

Parameter	Description
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 Sterling Integrator 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.
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.

Parameter	Description
Date	Extract attempt date. Format yyyyMMdd'T'hhmm. 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 Sterling Integrator 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 Sterling Integrator platforms
Related services	No
Application requirements	Nothing external to Sterling Integrator 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	Returned 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

	<p>Delete permission, or does not have permission to use the target mailbox.</p> <ul style="list-style-type: none"> • 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 Sterling Integrator 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 Sterling Integrator 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*.
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 Sterling Integrator 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.

Field	Description
	<ul style="list-style-type: none"> • 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 Sterling Integrator Mailbox before being deleted. You can use any number greater than zero. Required.
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	Pattern to select messages with names matching the pattern. The following characters cannot be used in a message name \ / : ? " < > .
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 <hr/> <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 Does this service start a business process? 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 value is any valid Sterling Integrator user ID.</p> <hr/> <p>Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select Admin 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 adapter will use the 24-hour clock instead of the default 12-hour clock.

Field	Description
Schedule	<p>Information about scheduling the business process invoked by this service. 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.
<p>Note: The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.</p>	

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
Messageld	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 Sterling Integrator 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 Sterling Integrator platforms
Related services	No
Application requirements	Nothing external to Sterling Integrator is required to use this service.
Initiates business processes?	No
Invocation	Not applicable (Internal service)
Business process context considerations	No
Returned status values	<p>Returned 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 – 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

	<p>contains specifications for more than one extractability type.</p> <ul style="list-style-type: none"> • 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 Sterling Integrator 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*.
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	<p>Indicates whether this message can be extracted. Valid values are Yes and No.</p> <hr/> <p>Note: You must provide a value for this field or one of the following:</p> <ul style="list-style-type: none"> • ExtractableCount • ExtractableUntil
ExtractableCount	<p>Number of times this message may be accessed. Optional. One of the three extractability parameters is returned. Valid values are ≥ 0.</p> <hr/> <p>Note: You must provide a value for this field or one of the following:</p> <ul style="list-style-type: none"> • Extractable • ExtractableUntil

Field	Description
ExtractableUntil	<p>The last date and time that this message may be extracted. Valid values are dates and times in <i>yyyyMMddThhmm</i> format.</p> <hr/> <p>Note: You must provide a value for this field or one of the following:</p> <hr/> <ul style="list-style-type: none"> • Extractable • ExtractableCount
MessageId	ID of the stored message. Required. Valid values are numbers ≥ 0 .
ResubmitMessage	Indicates if this message should be resubmitted for routing. Valid values are Yes and No.

Mapped Extraction Service

The Mapped Extraction service enables you to save data generated within Sterling Integrator 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 Sterling Integrator processing in a Visibility Services fact repository
Usage example	Business data that is generated during Sterling Integrator 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 Sterling Integrator.
Requires third party files?	No
Platform availability	All supported Sterling Integrator 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 Sterling Integrator 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	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>
```

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 Sterling Integrator 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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • 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>
    </operation>
  </sequence>
</process>
```



```

    </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 examples	Examples: <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 Sterling Integrator 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> and <i>Usage Examples</i> 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> and <i>Usage Examples</i> 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*.
 - 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*.

The XML representation of the MIME message contains three main elements that represent a MIME part. These elements include:

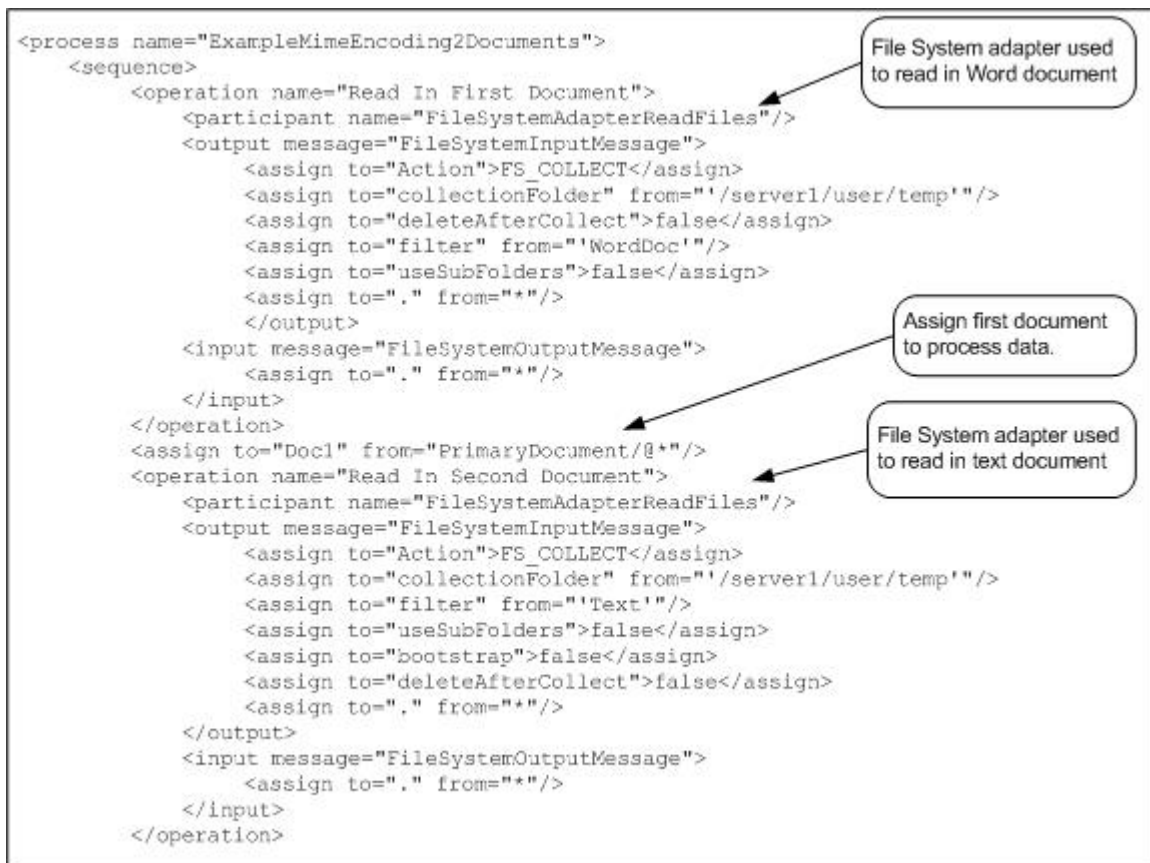
- mime:message – This element is the root node for the MIME part.
- mime:header – This element creates the specific header for the MIME part.
- mime:body – This element creates the content of the MIME part.

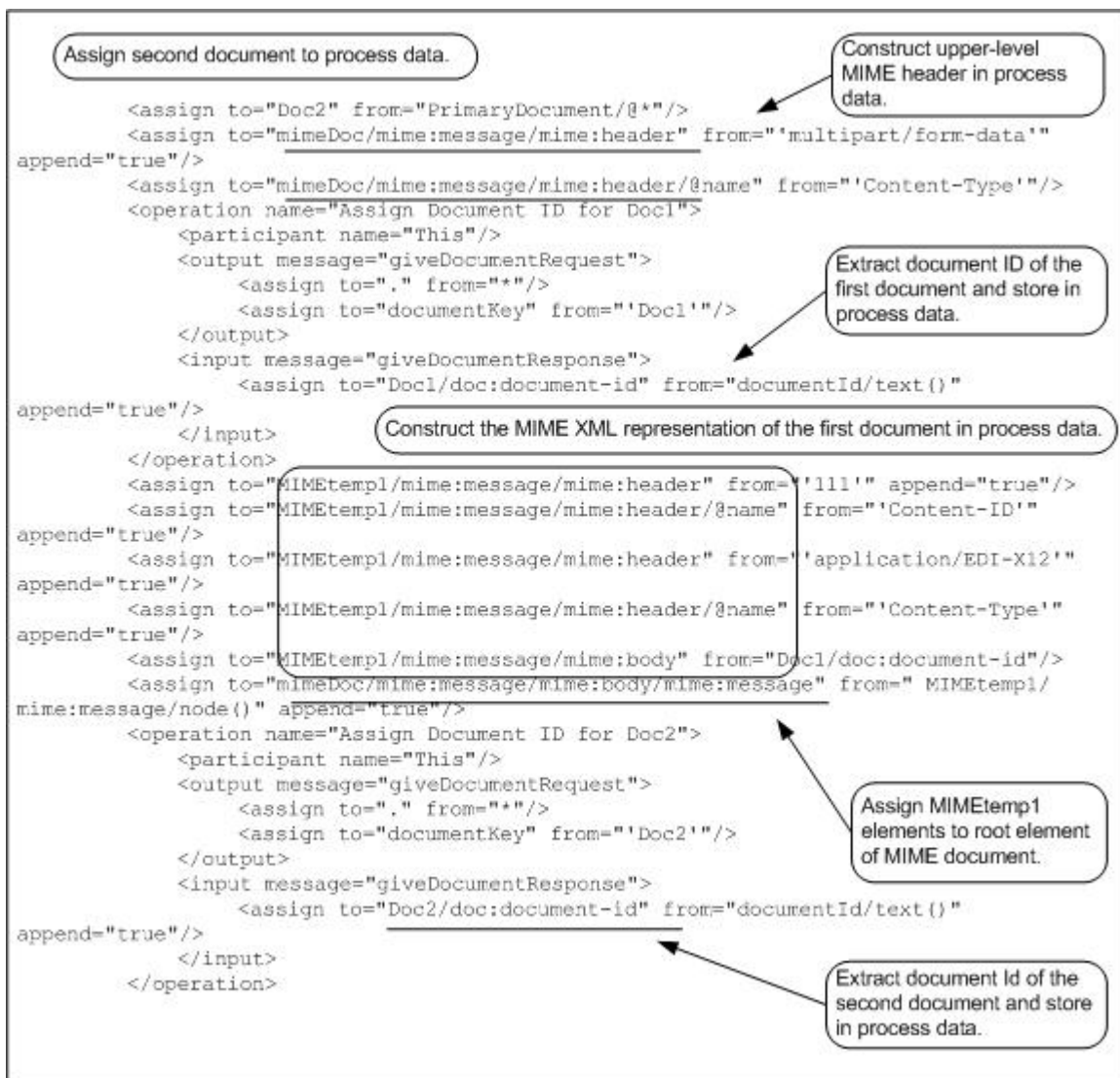
Because our example is a multipart message that contains two attachments, each MIME part has its own individual headers and body. To support this format, you need to create additional mime:message, mime:header, and mime:body elements underneath the first mime:body element for each of your attachments.

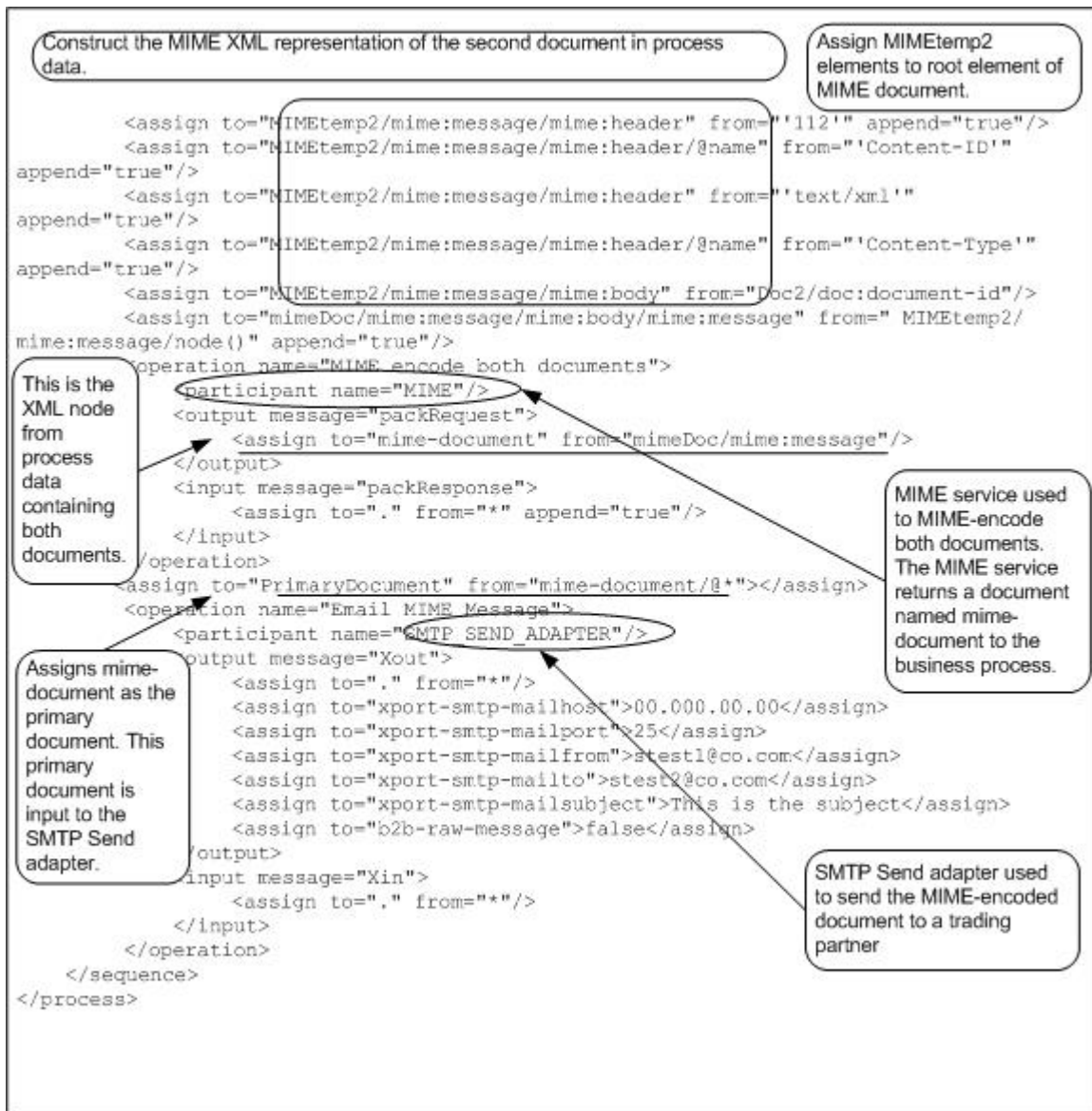
The example in the following section illustrates this point. In addition, see *Setting up Process Data for MIME Encoding*.

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:fbbb18bfda:-6b01</doc:document-id>
        </mime:body>
      </mime:message>
    </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.89g
-- serverName::76f6d6a8:fbbb18bfda:-6331::multipart.boundary.string
Content-ID: 112
Content-Type: text/xml

Hello World

-- serverName::76f6d6a8:fbbb18bfda:-6331::multipart.boundary.string

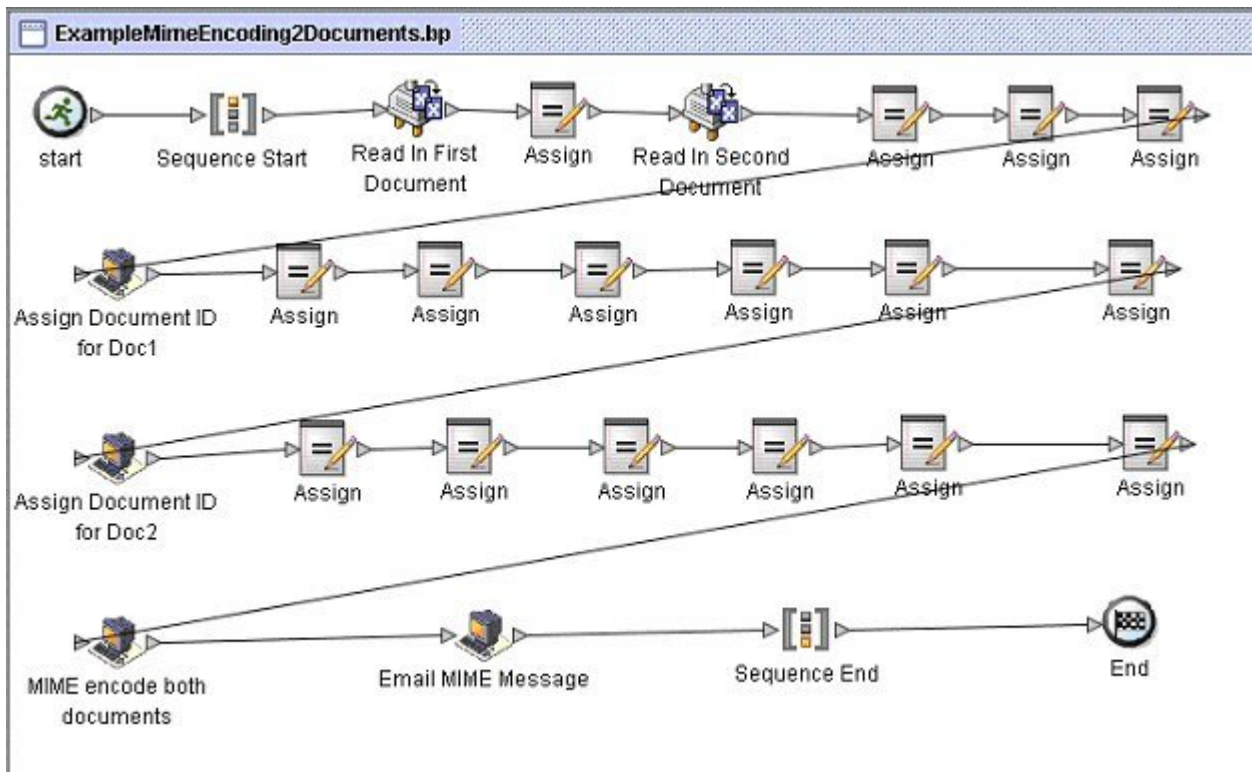
```

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*.

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* 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*.
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:

Service Editor - MIME encode documents

Name: MIME encode documents

Config: MIME

Message To Service: Message From Service

Output Msg: Messages Only

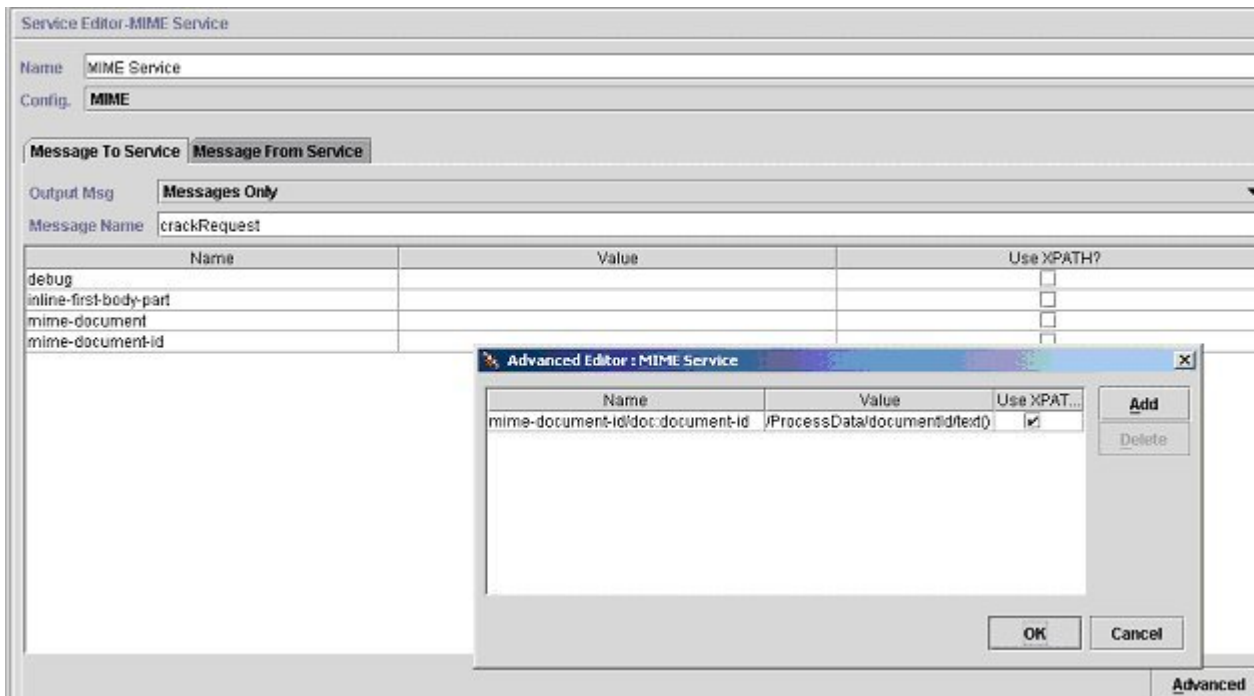
Message Name: packRequest

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.

Field	Description
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 crackRequest. Note: inline-first-body-part = true 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 packRequest.
mime-document-id	Node containing the document-id referencing the multipart document. Required when service starts with output message set to crackRequest.
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* 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:

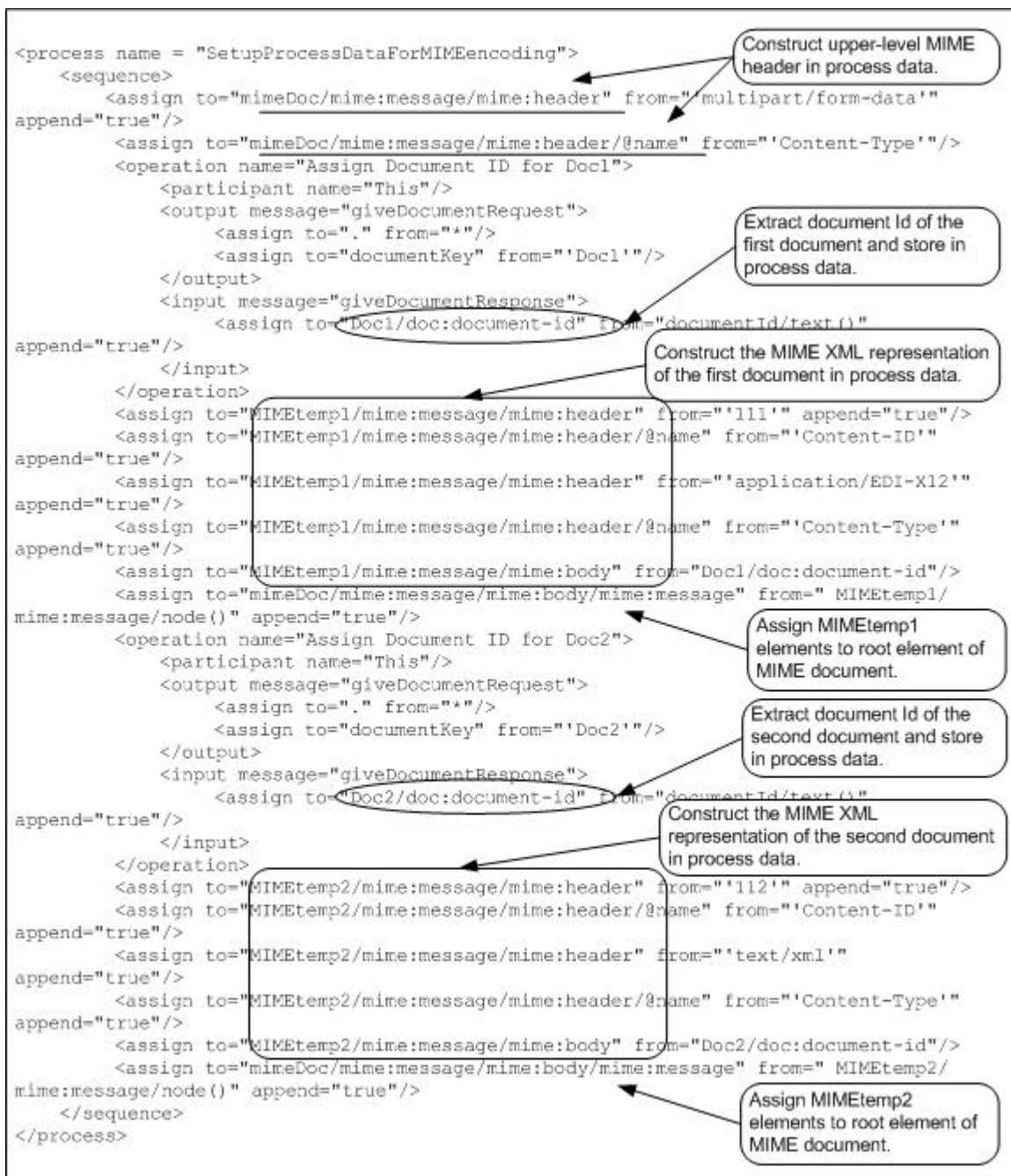
```
<mime:header name="Content-Transfer-Encoding">base64</mime:header>
```

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:fb4eddb774:-
5b83::multipart.boundary.string"
Content-Length: 19757

--serverName:5f6303:fb4eddb774:-5b83::multipart.boundary.string
Content-ID: 111
Content-Type: application/EDI-X12
_Microsoft Word Document
MSWordDoc_Word.Document.89q
--serverName:5f6303:fb4eddb774:-5b83::multipart.boundary.string
Content-ID: 112
Content-Type: text/xml

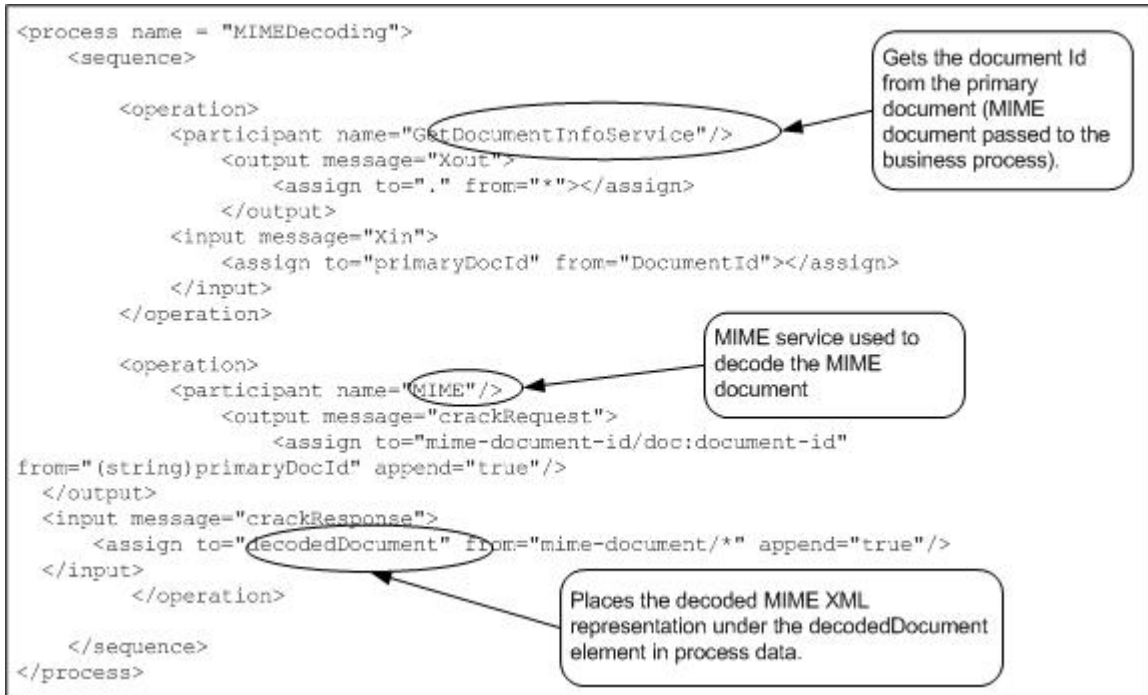
<Text>
</Text>
--serverName:5f6303:fb4eddb774:-5b83::multipart.boundary.string--

```

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.

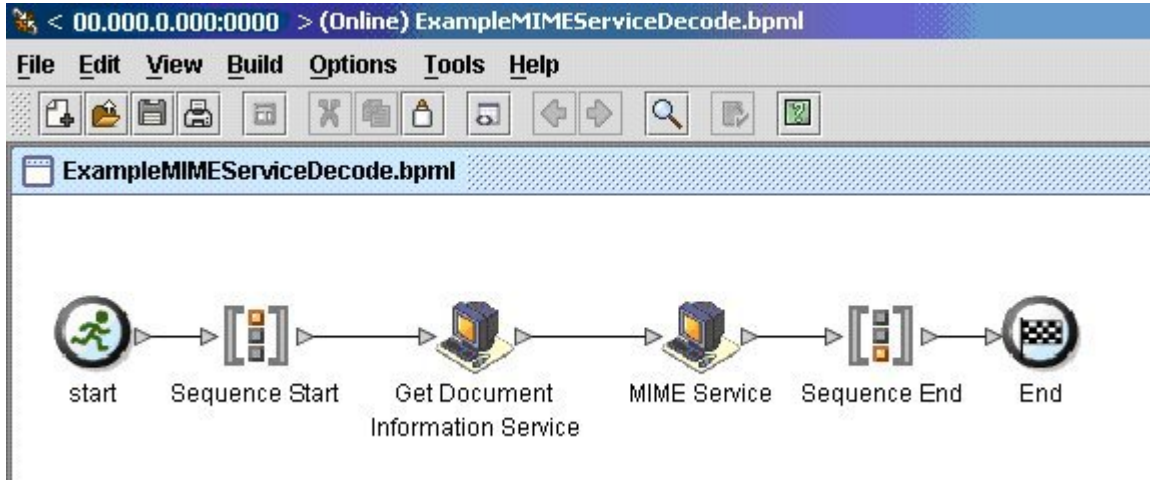
```

<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:fb4edbb774:-5a40
          </doc:document-id>
        </mime:body>

```

```
</mime:message>
  </mime:body>
</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.

```

<process name = "ExtractDocumentsFromMIME">
  <rule name="hasMoreDocuments">
    <condition> count(decodedDocument/mime:message/mime:body/mime:message) > 0 </
condition>
  </rule>
  <sequence>
    <assign to="index" from="0" />
    <choice name="GetDocuments">
      <select>
        <case ref="hasMoreDocuments" activity="ProcessDocuments"/>
      </select>
      <sequence name="ProcessDocuments">
        <assign to="index" from="index + 1" />
        <operation>
          <participant name="This"/>
          <output message="takeDocumentRequest">
            <assign to="." from="*" />
            <assign to="documentKey" from="concat('Document', index/
text())" />
            <assign to="documentId"
              from="decodedDocument/mime:message/mime:body/
mime:message[1]/ mime:body/doc:document-id/text()" />
            </output>
            <input message="takeDocumentResponse">
              <assign to="." from="*" />
            </input>
          </operation>
          <operation>
            <participant name="ReleaseService"/>
            <output message="releaseRequest">
              <assign to="TARGET" from="' decodedDocument/mime:message/
mime:body/mime:message[1]'" />
            </output>
            <input message="releaseResponse" />
          </operation>
          <repeat ref="GetDocuments" />
        </sequence>
      </choice>
    </sequence>
  </process>

```

Rule to loop through all the documents decoded by the MIME service

Used to rename the document

Release the document node in process data so the next document can be processed.

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 Sterling Integrator 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: <ProcessData> <RFH2> <mcd>....</mcd> <jms>....</jms> <usr>....</usr> <mqext>....</mqext> </RFH2> </ProcessData> 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: <mcd><Msd>none</Msd></mcd></p> <p>The following link contains information regarding the various folders:</p> <p>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><assign to="snd_MQMD_format" from=" `MQHRF2 ` "/></pre> • WebSphereMQ Suite (PutMessage): <pre><assign to="wsmq_MQMD_format" from=" `MQHRF2 ` "/></pre>
Returned status values	<p>Returned status values:</p> <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.
mqrfh2_msgFormat	Format name of the message payload data. Valid value is Any valid format type (see MQ manual). Default value is MQFMT_STRING. Optional.
mqrfh2_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" />
    </operation>
  </sequence>
</process>
```

```

    <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="'MQHRF2  '"/>
  </output>
  <input message="fromService"><assign to="." from="*" /></input>
</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

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 Sterling Integrator Admin UI.
Description	Sends messages to and reads messages from a remote Microsoft® Message Queue. You can configure the MSMQ adapter to send and retrieve messages: <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 Sterling Integrator. See <i>Implementing the MSMQ Adapter</i> .
Platform availability	All supported Sterling Integrator platforms
Related services	No

Application Requirements	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	Returned status values: <ul style="list-style-type: none"> • Success – The run was successful for the specified action • Error – An error occurred when the specified action was executed
Restrictions	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 Sterling Integrator for a SEND action. This only applies to transactional queues in MSMQ. • The MSMQ adapter does not support MSMQ acknowledgements.
Persistence Level	Full (default). <hr/> <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 Sterling Integrator 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 Sterling Integrator:

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*.
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 Sterling Integrator 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 all of the files in the `InstallJavaService` folder to the `MSMQ` folder so that `installwindowsservice`, `msmqproperties`, and the `Jacob` folder are in the same directory.
6. Copy `Jacob.dll` from the `msmqbundle*.jar/Jacob/1_7` folder to the `C:\WINDOWS\system32` folder.
7. Install the Java jdk version 1.5.0_11. Note the installation path.

8. 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`.

9. Change `MSMQ_SERVER_PORT` in `msmqprime.properties`, if necessary. The default is 8085. This is the port `msmqPrime` will run on.
10. 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.
11. Create a configuration of the MSMQ adapter in your Sterling Integrator and configure it to point to this `msmqPrime`.
12. Configure a service instance of MSMQ adapter in the Sterling Integrator to point to this `msmqPrime`.
13. Verify that the MSMQ adapter configuration is talking to this `msmqPrime` by including it in a busi-ness process and running it.
14. 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 debug option generates detailed logs.

Note: The `msmqbundle_xxx.jar` you use to create the `MSMQPrime` component must be from the same Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator and in the GPM.

Sterling Integrator Configuration

The following table describes the fields used to configure the MSMQ adapter in Sterling Integrator:

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.

Field	Description
	<ul style="list-style-type: none"> • 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.
EX_TRANSACTION	To include send action in the Sterling Integrator 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.

Field	Description
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'" />
      </output>
    </operation>
  </sequence>
</process>
```

```
<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. Sterling Integrator 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	Requirements: <ul style="list-style-type: none">• The connection from Application to the system where the MSMQ server resides.

	<ul style="list-style-type: none"> • 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	<p>Returned status values:</p> <ul style="list-style-type: none"> • Success - The run was successful for the specified action • Error - An error occurred when the specified action was executed
Restrictions	<p>Restrictions:</p> <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*.
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*.
4. Use the MSMQ Send service 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 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 Sterling Integrator 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 Send service needs to access it. It is recommended to convert it to an automatically started Windows service.
10. Create a configuration of the MSMQ Send service in your Sterling Integrator and configure it to point to this `msmqPrime`.
11. Configure a service instance of MSMQ Send service in the Sterling Integrator to point to this `msmqPrime`.
12. Verify that the MSMQ Send service 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 Sterling Integrator 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 Sterling Integrator 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. <hr/> 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.
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 Sterling Integrator 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_CORRELATIONID	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\testqueueetx</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>
      </output message>
    </operation>
  </sequence>
</process>
```

```
<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> <hr/> <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 Sterling Integrator. 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 Sterling Integrator Correlation Search function to view the network reconciliation status for EDI interchanges.
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • Success –The service completed successfully • Error –The service experienced a fatal error while processing
Restrictions	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. 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.
Persistence level	System default
Testing considerations	1) A business process that generates EDI data, sends it to the network, and calls this service to create the initial correlation records. 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.

How the Network Report Reconciliation Service Works

As part of the EDI enveloping process in Sterling Integrator, 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
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>
```

```
</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 - 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	In Sterling Integrator, 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. 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.
Usage example	Assume that you use Sterling Integrator 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 RevealDocument_Instance is available. There are no configuration parameters and no configuration is required.

	To use the service, just place RevealDocument_Instance in a business process.
Requires third party files?	No
Platform availability	All supported Sterling Integrator platforms
Related services	<p>This service is designed to work in conjunction with the Obscure Data - Obscure Primary Document service. 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> <hr/> <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 unobscure 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.

```
<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>
```

For more examples of obscuring and unobscuring the primary document, see the examples in *Obscure Data - Obscure Primary Document Service*.

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 Sterling Integrator, 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 Sterling Integrator 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 Sterling Integrator platforms
Related services	<p>This service works in conjunction with the Obscure Data - Reveal Primary Document service. 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> <hr/> <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/collect</assign>
    <assign to="extractionFolder">/dir_path/extract</assign>
    <assign to="unobscure">>true</assign>
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</process>
```

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 Sterling Integrator, 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 Sterling Integrator platforms

Related services	This service is designed to work in conjunction with the <code>RevealObscured XPath</code> function. See <i>revealObscured XPath Function</i> 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 Sterling Integrator, 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	Possible 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*.
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 Sterling Integrator:

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.

Field	Description
	<ul style="list-style-type: none"> • 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. <hr/> <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Parameter Name	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. This service creates process data elements based on the name-value pairs.
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 data-bbox="850 947 1472 1192"> <ProcessData> <admin>AAAAEQAAAAUAAAAeAAAAGgAAABkAAAA OAAAFgAAAK=</admin> <support>AAAAAAAAAAAAAAAAAAAAAAAAAAAA AAAAAAAA==</support> <ftpdemo>AAAFgAAABUAAAADAAAFwAAABIAA AAAHQAAAAI=</ftpdemo> </ProcessData> </pre> <p>This service creates process data elements based on the name-value pairs</p> <hr/> <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 http://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:

```
revealObscured (Name of Obscured Parameter)
```

Example:

```
revealObscured(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* 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)"append="true">

      </assign>

    <!-- Use the User ID and Password to gain access to a system folder -->
    INSERT OTHER BUSINESS PROCESS OPERATIONS HERE
  </sequence>
</process>
```

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, `User ID` 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.

```
<process name="Obscure_Data_Process_Data_Values_Example">
<sequence>
<!-- Use the Obscure Data - Process Data Values service to add obscured
    User ID and Password parameters 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 unobscured UserID and Password values from process data using
    the revealObscured XPath function -->
    <assign to="RevealedUserID" from="revealObscured(UserID)" append="true">
    </assign>
    <assign to="RevealedPassword" from="revealObscured>Password)" append="true">
    </assign>
    <!-- Use the unobscured UserID and Password values as needed in your business
        process -->
    INSERT OTHER BUSINESS PROCESS OPERATIONS HERE
    </sequence>
</process>
```

Odette FTP Adapter

The following table provides an overview of the Odette FTP adapter:

Note: Use the Odette FTP adapter instead of the OFTP adapter.

System name	Odette FTP Adapter
Graphical Process Modeler (GPM) category	All Services
Description	<p>This adapter is used to exchange files with trading partners using the Odette FTP (OFTP) protocol. There are two modes of operation:</p> <ul style="list-style-type: none"> • Queue Handler Mode • Manual Mode
Business usage	<p>The Odette FTP adapter supports point-to-point or VAN-like communication between trading partners using the OFTP protocol version 1.2, version 1.3, version 1.4, or version 2.0 over ISDN or TCP/IP as an underlying communication protocol. Version 2.0 supports a secure and authenticated communication between trading partners with the ability to encrypt and digitally sign message data, request signed receipts, and provide data compression. Trading partners can securely exchange business documents in two directions-- as the receiver or as the initiator of the document (in a push/pull fashion) while maintaining the integrity, authentication, and the origin of the document.</p>
Usage example	<p>Queue Handler Mode</p> <ul style="list-style-type: none"> • Initiator Role - The Odette FTP adapter is called from a business process with parameter PhysicalPartnerContract. All files or receipts in the Odette FTP Message Queue for this Physical Partner Contract with status "SCHEDULED" or "RETRY" are searched. An OFTP session is initiated and the related messages are sent to the remote partner. Optionally, the remote partner can send back messages for the initiator. • Responder Role - Incoming messages are stored in the root of the mailbox (optionally in submailboxes in the OFTP

	<p>Message Queue and if a Mailbox user is configured in the Physical Partner mailbox).</p> <p>Manual Mode</p> <ul style="list-style-type: none"> • Initiator Role: a set of documents can be sent to remote partner as specified in the data item xml structure. For each item in the data item xml structure, a new entry is created in the Odette FTP message queue. Then the Odette FTP adapter creates an Odette FTP session and sends all messages or receipts with status "SCHEDULED" or "RETRY" to the remote partner. Within the same session, the remote partner can send back files and receipts to the local partner. • Responder Role: the local partner accepts an incoming OFTP session from the remote partner and optionally, the remote partner sends files and receipts to the local partner. The remote partner then sends an OFTP change command. If the local partner has messages or receipts for the remote partner, then the local partner can send these messages in the same OFTP session (back to the remote partner). <p>For additional usage information about Mailbox and manual modes of operation, see Odette FTP Protocol Overview.</p>
Preconfigured?	<p>No.</p> <p>For manual mode, there is an Odette FTP 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 Odette FTP adapter implements the complete OFTP and CAPI/IP protocol stack.</p>
Platform availability	<p>All platforms supported by Sterling Integrator</p>
Related services	<p>Sterling Integrator Scheduler and Odette FTP Scheduler service Odette FTP Queue Handler service</p>
Application requirements	<p>For all modes and communication protocols:</p> <ol style="list-style-type: none"> 1. Create the Odette FTP partner profile. See Odette Partner Profile for additional information. 2. Create an Odette FTP 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. <p>For operation in Mailbox Mode:</p> <ol style="list-style-type: none"> 1. Create a Sterling Integrator User Account (Accounts > User Accounts). 2. Create a partner mailbox with a sub-mailbox named Inbox (Deployment > Mailboxes > Configuration).

	<p>3. Create a Virtual Root (Deployment >Mailboxes > Virtual Roots) for this mailbox.</p> <p>4. Enable the OFTPScheduler service.</p> <p>5. Use the Odette FTP Queue Handler to queue one or more files for sending.</p> <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 <code>oftpout</code> as a template for manual Mode.
Initiates business processes?	For each inbound message or receipt a business process is started. Create another business process <code>oftpin</code> for inbound files (Manual Mode).
Invocation	<p>Manual Mode: A business process initiating the Odette FTP Adapter manually or by a previous business process. Queue Handler Mode: The Odette FTP adapter is initiated by the Sterling Integrator Scheduler service based on a partner-specific schedule. It can also be initiated by inbound communication from a trading partner.</p>
Business process context considerations	<p>Queue Handler Mode: This is the recommended mode. The Queue Handler Mode can also be initiated manually. All messages are queued in the Odette FTP message queue in the partner mailbox. Manual Mode: Directly queued. The adapter takes one or more Primary Documents within the OFTPDataSet XML Input structure from the workflow.</p>
Returned status values	<p>Returned status values:</p> <ul style="list-style-type: none"> • 0 – Success • 1 – Error
Restrictions	<p>The Odette FTP adapter has the following restrictions:</p> <ul style="list-style-type: none"> • 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 Sterling Integrator Mailbox system requires a Mailbox license. • Special Logic is not supported. • Restarting is not supported. • Certificate Revocation Lists (CRL), introduced in OFTP 2.0, are not supported.
Persistence level	The adapter has no pre-set persistence level.
Testing considerations	For testing ISDN, you need a system handling the OFTP Partner role. For CAPI/ISDN communication, a BINTEC ISDN Router is required. In a loop back scenario, Sterling Integrator can be configured to take over the role of both an OFTP Initiator and an OFTP Responder.

Prerequisite

You must be familiar with the Odette File Transfer Protocol and ISDN or TCP/IP protocol.

Sample Business Scenario

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 Queue (Queue Handler Mode) or by passing more than one DataItem (document) in the OFTPDataSet XML structure (manual mode and Mailbox mode). This is much more efficient than initiating an OFTP session for each single file.

If the Odette FTP adapter is invoked, the business process remains in WAIT_ON_IO status until communication has ended. This includes sending and receiving files and acknowledgements (EERP/NERP). Then, the adapter reactivates the process and writes an OFTPResponse XML structure into the process data.

Note: The WAIT_ON_IO state avoids blocking business process thread resources when transferring data.

Implementing the Odette FTP Adapter in Queue Handler Mode

To implement the Odette FTP adapter in Queue Handler Mode:

1. If you want to use Mailbox Mode, create user accounts, virtual roots, and Mailboxes with Inboxes for you and all your partners.
2. Activate your license for the Odette FTP adapter and for the Mailbox (if used).
3. Create an Odette FTP adapter configuration.
4. Configure the adapter.

Note: Check the Odette FTP log file before using the adapter.

5. Create a business process that passes the PhysicalPartnerContract parameter to the adapter.
6. Test the business process and the adapter.
7. Enable the OFTP Scheduler service if you have created a Time Schedule in the Sterling Integrator Scheduler.
8. Use the Odette FTP Queue Handler service to send messages.

Implementing the Odette FTP Adapter in Manual Mode

To implement the Odette FTP adapter in manual mode:

1. Activate your license for the Odette FTP adapter.
2. Create an Odette FTP adapter configuration.
3. Configure the adapter.

Note: Check the Odette FTP log file before using the adapter. Verify that the adapter has been started without errors. Otherwise, the adapter is not ready to use.

4. Create and enable a business process that includes the Odette FTP adapter. You can use the sample business processes (oftpin and oftput) and modify them to suit your installation.
5. Create the OFTP Data Set XML structure in process data with the appropriate parameters so it can be passed to the adapter.
6. Test the business process and the adapter.

Configuring the Odette FTP Adapter

To configure the Odette FTP adapter, specify field settings in Sterling Integrator 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"> • ISDN • No Secure IP • Secure IP
Inbound Business Process Name	The name of the business process initiated for received files. Required for manual mode only. The business process name in the adapter configuration can be overridden by the business process selected in the Logical Partner Contract.
User	For manual mode, the Sterling Integrator 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 Sterling Integrator user ID. <p>Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select Admin as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p>

For Communication Mode of ISDN

Field	Description
ISDN Router Address	The name or IP address of the ISDN router. 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.

Field	Description
	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.

For Communication Mode of Non Secure IP

Field	Description
Perimeter Server	Name of the Perimeter Server or Perimeter Server group. Required.
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 Secure IP

Field	Description
Perimeter Server	Name of the Perimeter Server or Perimeter Server group. Required.
IP Server Listener Port	TCP/IP port number for incoming IP calls. Optional. Default is 6619.
Accept inbound calls without ACL check	Whether to verify inbound calls with the access control list. Default is Yes. Required.
System Certificate	Private key and certificate for server authentication. Required if Secure IP option is selected.
Cipher Strength	Strength of the algorithms used to encrypt data. Valid values are: <ul style="list-style-type: none"> • ALL

Field	Description
	<ul style="list-style-type: none"> • WEAK • STRONG - Default.
CA Certificate	Select one or more CA Certificates. Certificate used to validate the certificate of an OFTP client. This is the public key. If no CA certificate is selected, no client certification is performed. Optional.

Example Business Process for Queue Handler Mode

This example shows a business process that sends one message to a partner:

```

<!-- Adapter: Odette FTP
Description: Example template process for initiating an OFTP session for a
           Physical Partner Contract
           (Odette FTP Queue Handler Mode)
Prerequisites:
- Read the Odette FTP documentation
- For communication type "ISDN" setup your ISDN Router. For communication type
  "IP", the GIS Perimeter Service is used.
- Configure the OFTP Partner Profile (use OFTP Partner Profile User Interface
  or edit the PartnerProfile XML file template <gis_root>/install/properties/
  PartnerProfile.xml and use script PartnerManager.sh to import the Partner
  Profile into the database).
- Configure and enable the Odette FTP Adapter instance "OFTPSendFile"
- Put files into the Odette FTP Message Queue using the Odette FTP Queue Handler
  (use template bpml "oftpfile")
- Create time schedules for initiating OFTP sessions for with the GIS Scheduler.

  Enable the GIS Scheduler and OFTP Scheduler Service.
Input : <PhysicalPartnerContract/>ppc_name</PhysicalPartnerContract/>, where
       ppc_name is the name of the Physical Partner Contract used by the Odette
       FTP Adapter to search all entries in the OFTP mMessage Queue for this PPC
       with status "SCHEDULED".
-->
<process name="oftpinitsession">
  <sequence name="oftp">
    <!-- Start OFTP send process in Queue Handler Mode-->
    <operation name="SendOFTP">
      <participant name="OFTPSendFile"/>
      <output message="Out">
        <assign to="." from="*"></assign>
      </output>
      <input message="In">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>

```

Parameter Passed from Business Process to Adapter - Queue Handler 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 Odette FTP 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 Odette FTP log for error details. The Partner Profile configuration may have errors.
WRONG_WF_PARAM	In Queue Handler 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 (**oftpout** for initiating a session to a partner) that you can modify to use with the Odette FTP 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 Odette FTP 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 Odette FTP adapter is able to send one or multiple Sterling Integrator 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:

```
<OFTPDataSet PhysicalPartnerContract="physical_partner_contract_name">
  <DataItem_1>
    <properties>
```

Required.

```
<LogicalPartnerContract>log_partner_contract_name</LogicalPartnerContract>
```

Optional. 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>
<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, refer to the table below. This table describes the parameters that need to match the definitions in the OFTP partner profile. Verify 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@PhysicalPartnerContract	The unique name of the physical partner contract as defined in the Partner Profile. This is an 80-character string. Required.
LogicalPartnerContract	The unique name of the logical partner contract as defined in the Partner Profile. 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 and higher) Optional.

Parameter	Description																									
	<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 and higher) <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>																									
FileFormat	<p>This field specifies the format of the virtual file. Valid values are:U - unstructured binary fileT - text fileF - fixed-length record binary fileV - variable-length record binary fileOptional.</p>																									
OFTPFileUserField	<p>Used as defined by your bilateral agreement with your partner. Optional.</p>																									
RecordDelimiter	<ul style="list-style-type: none"> • Depending on File Format: One or two record delimiters (decimal numbers of Character code, for example, 13,10 for <CR>>LF> (Windows) or 10 for <LF> (UNIX). • File Format "Unstructured Binary file" ("U") / "Text" ("T"): • Data is not split up in records. Delimiters are not used. In the OFTP Partner Profile database both Record Delimiters are set to (-1,-1). • File Format "Variable format binary file" ("V"): • Data is split up in records separated by one or optionally two record delimiters. Specify one or two delimiters as decimal values. Specify -1,-1 to use Operating system dependent default delimiter (13,10 for Windows and 10 for UNIX) • File Format "Fixed Format binary file": • Data is split up in records of length <p>For Format "F" there are specified special cases:</p> <table border="1"> <thead> <tr> <th>Format</th> <th>RecLen</th> <th>Del 1</th> <th>Del 2</th> <th>Sending Party</th> </tr> </thead> <tbody> <tr> <td>F</td> <td><rec. len></td> <td>-2</td> <td>-2</td> <td>Split data in rec. len blocks</td> </tr> <tr> <td>F</td> <td><rec. len></td> <td><d1></td> <td>-1</td> <td>removed d1 from stream</td> </tr> <tr> <td>F</td> <td><rec. len></td> <td><d1></td> <td><d2></td> <td>removed d1/d2 from stream</td> </tr> <tr> <td>F</td> <td><rec. len></td> <td>-1</td> <td>-1</td> <td>Split data in rec. len blocks</td> </tr> </tbody> </table>	Format	RecLen	Del 1	Del 2	Sending Party	F	<rec. len>	-2	-2	Split data in rec. len blocks	F	<rec. len>	<d1>	-1	removed d1 from stream	F	<rec. len>	<d1>	<d2>	removed d1/d2 from stream	F	<rec. len>	-1	-1	Split data in rec. len blocks
Format	RecLen	Del 1	Del 2	Sending Party																						
F	<rec. len>	-2	-2	Split data in rec. len blocks																						
F	<rec. len>	<d1>	-1	removed d1 from stream																						
F	<rec. len>	<d1>	<d2>	removed d1/d2 from stream																						
F	<rec. len>	-1	-1	Split data in rec. len blocks																						

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 Odette FTP adapter call. The following example shows a sample response structure:

```
<OFTPResponse PhysicalPartnerContract="physical_contract_name">
  <DataItem_1>
    <Status>[Success|Failure|Skipped]</Status>
    <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

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. <hr/> 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.
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 and higher) <hr/> Note: The virtual file name, date, and time are used to uniquely define a file.

Parameter	Description
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 and higher) <p>Note: The virtual file name, date, and time are used to uniquely define a file.</p>
Originator	Identifies the sender of the virtual file. This is the location that mapped the data for transmission.
Destination	Identifies the final recipient of the transmission. This is the location that looks into the virtual file content.
FileFormat	<p>This field specifies the format of the virtual file. Valid values are:</p> <ul style="list-style-type: none"> • 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 Sterling Integrator 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="*" />
      </output>
      <input message="toProcessData">
        <assign to="." from="*" />
      </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>
```

Odette FTP Queue Handler Service

The following table provides an overview of the Odette FTP Queue Handler service:

System name	Odette FTP Queue Handler Service
Graphical Process Modeler (GPM) category	All Services
Description	The Odette FTP Queue Handler is used to queue a file for sending by adding entries to the process queue and to an associated Mailbox, if used. These entries contain all information required by the Odette FTP adapter to identify the Physical Partner Contract and prepare a file for sending, for example, the Sterling Integrator document or MessageID, Logical Partner Contract, and OFTP file name. In OFTP 2.0, a file can be compressed, encrypted and signed in a CMS enveloped before it is persisted into the Odette FTP Message Queue. In DECIPHER Mode, an enveloped file in the Odette FTP Message Queue can be decrypted and written into the process data for further processing (same applies for compressed and/or signed files).
Business usage	Use the Odette FTP Queue Handler to put a file into the send queue for a partner. The 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. In DECIPHER Mode, an encrypted file in the Odette FTP Message Queue can be decrypted and written into the process data for further processing (the same applies for compressed and/or signed files).
Usage example	There is a preconfigured instance of the Odette FTP Queue Handler service named OFTPQueueHandler. Three business process templates are provided to put items in the send queue: <ul style="list-style-type: none"> • oftpfile: Put a file into the OFTP queue • oftppeerp: 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 Sterling Integrator
Related services	Odette FTP Scheduler service Odette FTP adapter (Sterling Integrator Scheduler)
Application requirements	The Odette FTP Queue Handler service is used in conjunction with the Odette FTP adapter in the Sterling Integrator Scheduler.
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	<p>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. Optionally, if Mailbox mode is used, 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 Odette FTP adapter to identify the related OFTP send parameters defined in the Physical Partner Contract.</p>
Returned status values	<p>Possible values:</p> <ul style="list-style-type: none"> • 0 – Success • 1 – Error
Restrictions	None
Persistence level	System default
Testing considerations	Not applicable
mode	<p>The WFC parameter "mode" determines whether the Odette FTP Queue Handler works in "QUEUE" mode or "DECIPHER" mode. Valid values:</p> <ul style="list-style-type: none"> • DECIPHER: used for offline processing of a received message, such as removing CMS envelope, message signature and/or decrypting an encrypted message. The resulting document is written back into the process data. • QUEUE (default): puts a message in the Odette FTP Message Queue for sending a message. Optionally, in OFTP 2.0 a file can be compressed, encrypted or signed. <p>For both QUEUE and DECIPHER modes, you can either:</p>

- Specify the Mailbox message Id of the message (new WFC-Parameter "MessageId") or
- Pass a Primary document in process data (such as use the business process which is started for each received file together with a Primary Document in the process data).

Configuring the Odette FTP Queue Handler Service in Queued Mode

There are no configurable service instance parameters for the Odette FTP Queue Handler service.

If Odette FTP version 2.0 is used, the OdetteFTP Queue Handler service is optionally responsible for signing, encrypting and compressing files before queueing them.

To use the Odette FTP Queue Handler service:

1. Create an Odette FTP Partner Profile.
2. Create the following XML structures in a business process.
3. Pass the XML structures to the Odette FTP Queue Handler service from the business process.

Input XML Structures

To put files, EERPs or NERPs into the OFTP queue, the Odette FTP 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:

```
<OFTPDataItem>
  <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_
      <PrimaryDocumentSCIOBJECTID="document_id_1"/>
    </document>
  </FILE>
</OFTPDataItem>
```

Defining XML Node Name Parameters

To define the XML node name parameters:

1. Make the following changes to the XML structure.

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. 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 and higher) 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 and higher) Optional, but it is recommended that you specify the date and time stamp when the application created the file. 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. 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	Depending on File Format: One or two record delimiters (decimal numbers of Character code, for example, 13,10 for <CR><LF> (Windows) or 10 for <LF> (UNIX). <ul style="list-style-type: none"> • File Format " Unstructured Binary file" ("U") / "text" ("T"):

Parameter	Description																														
	<p>Data is not split up in records. Delimiters are not used. In the OFTP Partner Profile database, both Record Delimiters are set to (-1,-1).</p> <ul style="list-style-type: none"> File Format "Variable format binary file" ("V"): <p>Data is split up in records separated by one or optionally two record delimiters.</p> <p>Specify one or two delimiters as decimal values. Specify -1,-1 to use Operating system dependent default delimiter (13,10 for Windows and 10 for UNIX)</p> File Format "Fixed Format binary file": <p>Data is split up in records of length.</p> <p>For Format "F" there are following specified cases:</p> <table border="1"> <thead> <tr> <th>Format</th> <th>Rec Len</th> <th>Del 1</th> <th>Del 2</th> <th>Sending Party Specified</th> <th>Receiving Party</th> </tr> </thead> <tbody> <tr> <td>F</td> <td><rec_len></td> <td>-2</td> <td>-2</td> <td>Split data in rec_len blocks</td> <td>appends OS specific delims (1310 or 10)</td> </tr> <tr> <td>F</td> <td><rec_len></td> <td><d1></td> <td>-1</td> <td>removed d1 from stream</td> <td>appends <d1> to output data stream</td> </tr> <tr> <td>F</td> <td><rec_len></td> <td><d1></td> <td><d2></td> <td>removed d1&d2 from stream</td> <td>appends <d1> and <d2> to out_data stream</td> </tr> <tr> <td>F</td> <td><rec_len></td> <td>-1</td> <td>-1</td> <td>Split data in rec_len blocks</td> <td>Do NOT append anydelims to out_Stream (new default)</td> </tr> </tbody> </table> 	Format	Rec Len	Del 1	Del 2	Sending Party Specified	Receiving Party	F	<rec_len>	-2	-2	Split data in rec_len blocks	appends OS specific delims (1310 or 10)	F	<rec_len>	<d1>	-1	removed d1 from stream	appends <d1> to output data stream	F	<rec_len>	<d1>	<d2>	removed d1&d2 from stream	appends <d1> and <d2> to out_data stream	F	<rec_len>	-1	-1	Split data in rec_len blocks	Do NOT append anydelims to out_Stream (new default)
Format	Rec Len	Del 1	Del 2	Sending Party Specified	Receiving Party																										
F	<rec_len>	-2	-2	Split data in rec_len blocks	appends OS specific delims (1310 or 10)																										
F	<rec_len>	<d1>	-1	removed d1 from stream	appends <d1> to output data stream																										
F	<rec_len>	<d1>	<d2>	removed d1&d2 from stream	appends <d1> and <d2> to out_data stream																										
F	<rec_len>	-1	-1	Split data in rec_len blocks	Do NOT append anydelims to out_Stream (new default)																										
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 Sterling Integrator 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:

```
<OFTPDataItem>
  <EERP>
    <properties>
      <LogicalPartnerContract>log_partner_name</LogicalPartnerContract>
      <OFTPVirtualFilename>virtual_filename</OFTPVirtualFilename>
      <Date>date</Date>
      <Time>time</Time>
    </properties>
  </EERP>
</OFTPDataItem>
```

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.

Parameter	Description
OFTPVirtualFilename	<p>The OFTP Virtual File name. Defined according to the bilateral agreement with your trading partner. This is a 26-character string. Required. If omitted a default file name is taken from the partner profile contract.</p> <hr/> <p>Note: The virtual file name, date, and time are used to uniquely define a file.</p>
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 and higher) <p>Optional.</p> <hr/> <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 and higher) <p>Optional, but it is recommended that you specify the date and time stamp when the application created the file.</p> <hr/> <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:

```

<OFTPDataItem>
  <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>
</OFTPDataItem>

```

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.

Parameter	Description
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	The date tag used to send the message. Required. 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. Required. 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.
Reason	Reason code for the NERP. Required. Valid values are two-digit numbers defined in the OFTP 2.0 specification.

Decipher Mode

To decrypt an encrypted OFTP file in a Primary Document or Mailbox (specified by Message Id) the Odette FTP Queue Handler has to be called in DECIPHER mode and/or compressed or signed. The following example shows the usage of the Queue Handler for this purpose:

```
<!--
Adapter      : OdetteFTP Queue Handler - DECIPHER mode
Description  : Example template process for decrypting a received encrypted
               file. This process can be used in the business process
               which is started for inbound files.
Prerequisites: - Read the OdetteFTP Queue Handler documentation

               - Configure and enable the OdetteFTP Adapter instance
                 "OFTPQueueHandler"
               - Modify this template process as described in the
                 comments below.
               - Make sure OFTP Partner Profile has been imported
                 into the database.

Input : Parameter
       mode          DECIPHER Either      MessageId      <number>
       Queue Handler searches for an file entry in the Odette FTP Queue
       with Message ID <number>. If there is a file it is decrypted. or
```

```

        PPrimaryDocument.    Pass PrimaryDocument which needs to be decrypted
        in the process data. -->
<process name="oftpdecrypt">
  <sequence name="decrypt">
    <!--
      Either pass parameters from an initiating process and remove the assign
      statements _or_ enter the values in the following assign statements.
    -->

    <assign to="mode" from="'DECIPHER'" />
  <!--
    <assign to="MessageId">number</assign>
  -->
  <!-- Decipher document -->
  <operation name="DecryptFile">
    <participant name="OFTPQueueHandler" />
    <output message="OFTPOut">
      <assign to="." from="*" />
    </output>
    <input message="OFTPIn">
      <assign to="." from="*" />
    </input>
  </operation>
</sequence>
</process>

```

In Manual Mode the Odette FTP Queue Handler could be called in DECIPHER mode directly from the business process initiated for an encrypted inbound file.

In Queued Mode an Automated Mailbox rule could be used to initiate a business process for inbound files calling the Odette FTP Queue Handler in DECIPHER mode.

Error Messages (both Queued and Decipher modes)

The following error messages will be only displayed in the Status Report for the instance of the Odette FTP 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 Odette FTP.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 Odette FTP.log for details.
Odette FTPQueueHandler_FAILURE	A general error occurred during processing that does not fit into the categories listed above. The reason is noted in the status report/Odette FTP.log.

Error Codes for Decipher Mode

Error Code	Description
NO_PD_FOUND	DECIPHER mode. Processed data where no primary document could be found. Pass a primary document to the service.
NO_ENCRYPTED_FILE_FOUND	DECIPHER mode. Either a primary document is found in process data, but an entry with document ID could not be found in the FTP_OBJECT table, or there is no entry in the OFTP_OBJECT table with the Message Id passed to the service.
WRONG_OFTP_OBJECT_VERSION	DECIPHER mode. The file in the OFTP_OBJECT table is not received with the required OFTP Version 2.0.
DOCUMENT_NOT_ENVELOPED	DECIPHER mode. A file in the OFTP_OBJECT table is found but it is has no CMS envelope. Parameter SFIDENV in OFTP_OBJECT_EXT is not "1").

OdetteFTP Scheduler Service

The following table provides an overview of the service:

System name	Service
Graphical Process Modeler (GPM) category	All Services
Description	The 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 service initiates the OdetteFTP adapter (if the partner contract defines Mailbox Mode) based on a partner specific schedule.
Usage example	<p>The 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 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 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 Sterling Integrator

Related services	OdetteFTP adapter (Mailbox mode only), OdetteFTP Queue Handler service
Application requirements	The 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 Service

This service is not called by a business process. It is started automatically when is started and shuts down when is stopped. Enable and disable the service by starting and stopping the service in the Administration menu.

Configuring the Service

To configure the service, you must specify field settings in Sterling Integrator 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 Sterling Integrator.

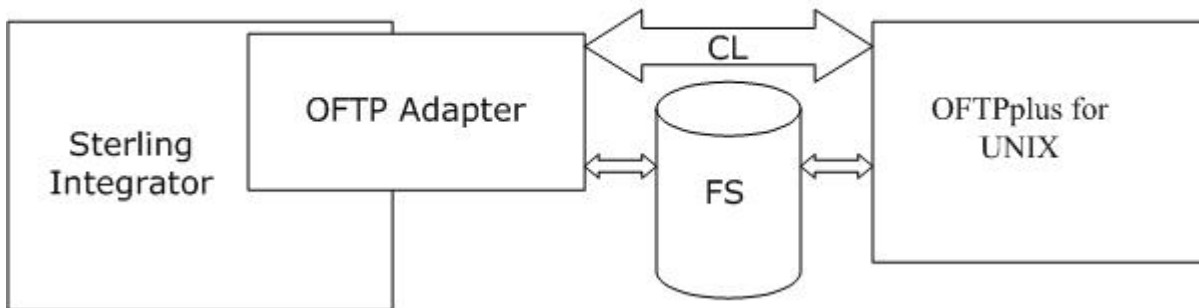
System name	Oftp Adapter
Graphical Process Modeler (GPM) category	None
Description	This adapter is used to exchange files with trading partners via the Odette FTP (Oftp) protocol.
Business usage	The Oftp Adapter uses OFTPplus for UNIX to support point-to-point communication between trading partners using 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	Available for: <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	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 Sterling Integrator. 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.
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

Mode	Wait	Behavior
		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, Oftpys 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 Oftpys 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*.
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 Sterling Integrator:

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	<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>
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.
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)	<p>Directs handling of the file for processing. Required. Valid values:</p> <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

Field	Description
	<ul style="list-style-type: none"> • Send immediately without waiting for response – Synchronous mode with no wait • Send immediately and wait for response – Synchronous mode with wait <p>See <i>Oftp Adapter Modes</i>.</p>
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.
User	<p>User ID to use with the business process. Type the user ID, or click the  icon and select a user ID from the list. Valid values:</p> <ul style="list-style-type: none"> • Any valid Sterling Integrator user ID <hr/> <p>Note: This parameter allows someone who doesn't have rights to a specific business process to run it. If you select Admin as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run.</p> <hr/>

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.

Field	Description
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 Sterling Integrator.
Preconfigured?	No
Requires third party files?	Yes. aqapi.jar, classes12.zip
Platform availability	All supported Sterling Integrator 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> .
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. Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator not responding for a long time, break the connections on the AQ server to force AQ to close connections with Sterling Integrator and release the threads. If this is a consistent problem in your Oracle environment, review the Sterling Integrator 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 Sterling Integrator and Oracle installed.
3. If it is running, stop Sterling Integrator.
4. From the bin directory where Sterling Integrator is installed, install the necessary third party files (aqapi.jar and classes12.zip) using the install3rdParty script included with Sterling Integrator. 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 Sterling Integrator.
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*.
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 Sterling Integrator:

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. <hr/> <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.
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> .

Field	Description
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 Sterling Integrator.
Preconfigured?	No
Requires third party files?	Yes. aqapi.jar, classes12.zip
Platform availability	All supported Sterling Integrator 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> .
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. Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator and Oracle installed.
3. If it is running, stop Sterling Integrator.
4. From the bin directory where Sterling Integrator is installed, install the necessary third party files (aqapi.jar and classes12.zip) using the install3rdParty script included with Sterling Integrator. 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 Sterling Integrator.
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*.
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 Sterling Integrator:

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 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.

Field	Description
	<ul style="list-style-type: none"> • 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. <hr/> <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)	Type of topic to access. Valid values are: <ul style="list-style-type: none"> • Topic Send – Send messages. • Topic Receive Sync – Receive messages.
Message Type(payload)	Type of message to send. Valid values are: <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.
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.

Field	Description
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 Sterling Integrator.
Usage example	<p>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:</p> <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 Sterling Integrator 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	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*.
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 Sterling Integrator 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 Sterling Integrator:

Field Name	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.
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 CONTROLAREA 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 CONTROLAREA 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)

Field Name	Description
	<ul style="list-style-type: none"> • 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.
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.

Field Name	Description
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.
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)	<p>The OTA protocol REQUEST_TYPE parameter. The default value SEND should be used for normal transactions. Default is SEND. Required. Valid values are:</p> <ul style="list-style-type: none"> • SEND • AUTH • AUTH2 • EME <p>For more information about these values, see your Oracle documentation.</p>
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	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. Oracle sends an invoice to Sterling Integrator. 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.
Preconfigured?	No
Requires third party files?	OAG DTDs must be checked in to Sterling Integrator for each document type exchanged with Oracle.
Platform availability	All supported Sterling Integrator 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	Possible values are: <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*.
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.

Field	Description
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.
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:

<p>PrimaryDocument:</p> <ul style="list-style-type: none"> • 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	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. 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.
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?	No. You must create a trading partner contract, a perimeter server, and configurations of these services or adapters: <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 Sterling Integrator for each document type exchanged with Oracle.
Platform availability	All Sterling Integrator supported platforms

<p>Related services</p>	<p>Related services include:</p> <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 Sterling Integrator 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
<p>Application requirements</p>	<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)
<p>Initiates business processes?</p>	<p>The HTTP Server adapter must be configured with a URL that runs the OracleEBusinessReceive business process when a message is received from Oracle.</p>
<p>Invocation</p>	<p>To send a message to Oracle, create a business process that does the following: Starts the Oracle E-Business Suite Adapter Configuration service. Places an OAG XML document (for example, PROCESS PO) into the primary document . Runs the OracleEBusinessSend business process . To receive a message from Oracle, do the following: Configure a URL in the HTTP Server adapter that starts the OracleEBusinessReceive . 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). Use the Oracle E-Business Suite Adapter Configuration service to configure the option you select.</p>
<p>Business process context considerations</p>	<p>See Invocation.</p>
<p>Returned status values</p>	<p>Returned status values:</p> <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.
<p>Restrictions</p>	<p>Oracle E-Business Suite 12.0.0 and the adapter both support only UTF-8 encoding.</p>

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*
- *Receiving Documents from Oracle*

The adapter uses a customized configuration of the BP Fault Log adapter called OracleEBusinessAdapterLogger to log all errors.

Send 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 Sterling Integrator.

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

4. The Oracle E-Business Suite adapter constructs and encodes the message using the Oracle E-Business Suite Message service.
5. 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.
6. 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.

Receive Documents from Oracle

1. The Oracle E-Business Suite sends an OAG XML document to Sterling Integrator. The HTTP Server adapter must have a URI configured that matches the URI where the document was posted and runs the OracleEBusinessReceive business process.
2. The OracleEBusinessReceive business process sends back a synchronous HTTP response to Oracle to close the HTTP connection.
3. 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.

4. The adapter decodes the message using the Oracle E-Business Suite Message service and puts the OAG XML payload into the primary document.
5. 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 Sterling Integrator.

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

6. 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 Sterling Integrator.

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.

Implement 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*.
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 Sterling Integrator 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 Sterling Integrator. 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 Sterling Integrator and Oracle:
 - a. Sterling Integrator 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*.

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* 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*.
10. In Sterling Integrator, 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*.
- Oracle E-Business Suite Message service: for more information, see *Oracle E-Business Suite Message Service*.
- Timezone Offset service: for more information, see *Timezone Offset Service*.

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	<p>The top-level business process to send a message to Oracle.</p> <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 OracleEBusinessInsertDate. 3. Run the business process OracleEBusinessCorrelate. 4. Encode the XML document using the Oracle E-Business Suite Message service. 5. Send the message to Oracle E-Business Suite. 6. If a Confirm BOD is expected, wait for it.
OracleEBusinessInsertDate	<p>Replace the values within the CNTROLAREA/DATETIME element of the primary document with the current time.</p>
OracleEBusinessCorrelate	<p>Uses the Correlation service to correlate the primary document. See the correlation table in the section <i>Sending Documents to Oracle</i> for a list of the values correlated.</p>
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>

Edit 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 Sterling Integrator. 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 Sterling Integrator 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 Sterling Integrator certificate (B2BHttp).

If you are using HTTP:

- For **SSL**, select **None**.
- All three certificate fields can be left blank.

Create 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) – Sterling Integrator sends data to PeopleSoft. No response from PeopleSoft is necessary.
- PeopleSoft Send and Wait (PS Send and Wait) – Sterling Integrator sends a request to PeopleSoft, receives a response from PeopleSoft, and ties the response back to the waiting business process.
- PeopleSoft Receive (PS Receive) – Sterling Integrator 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. It also 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	Set of business processes used within Sterling Integrator 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 Sterling Integrator 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. 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.

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. 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. 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> <hr/> <p>Note: DTDs and schemas are available from PeopleSoft to meet mapping needs.</p>
Platform availability	All supported Sterling Integrator platforms
Related services	<p>Related services include:</p> <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

	The adapter also requires a Perimeter server.
Application requirements	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.
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 Sterling Integrator 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 Sterling Integrator Map Editor.
- Create and edit a business process using the Sterling Integrator GPM.

For the Adapter for PeopleSoft to work correctly, you must establish an HTTP connection to PeopleSoft from Sterling Integrator 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.

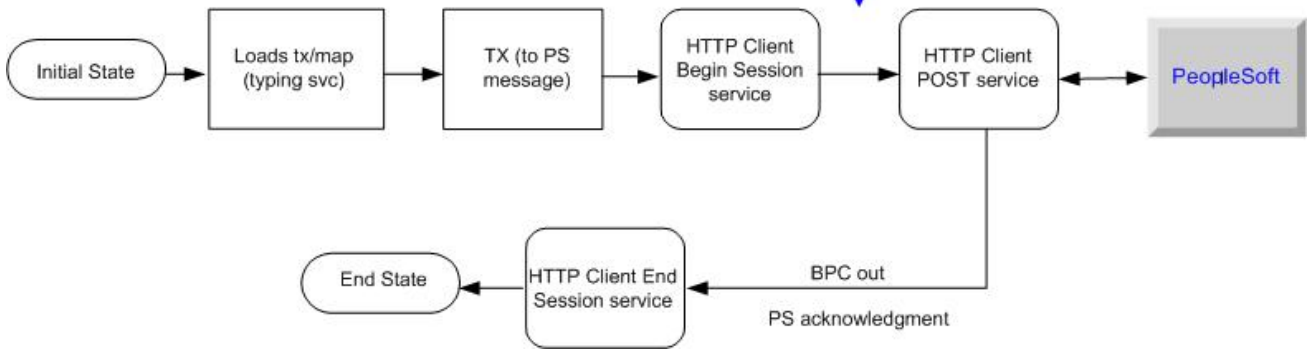
PeopleSoft Send Business Process (PS Send)

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:



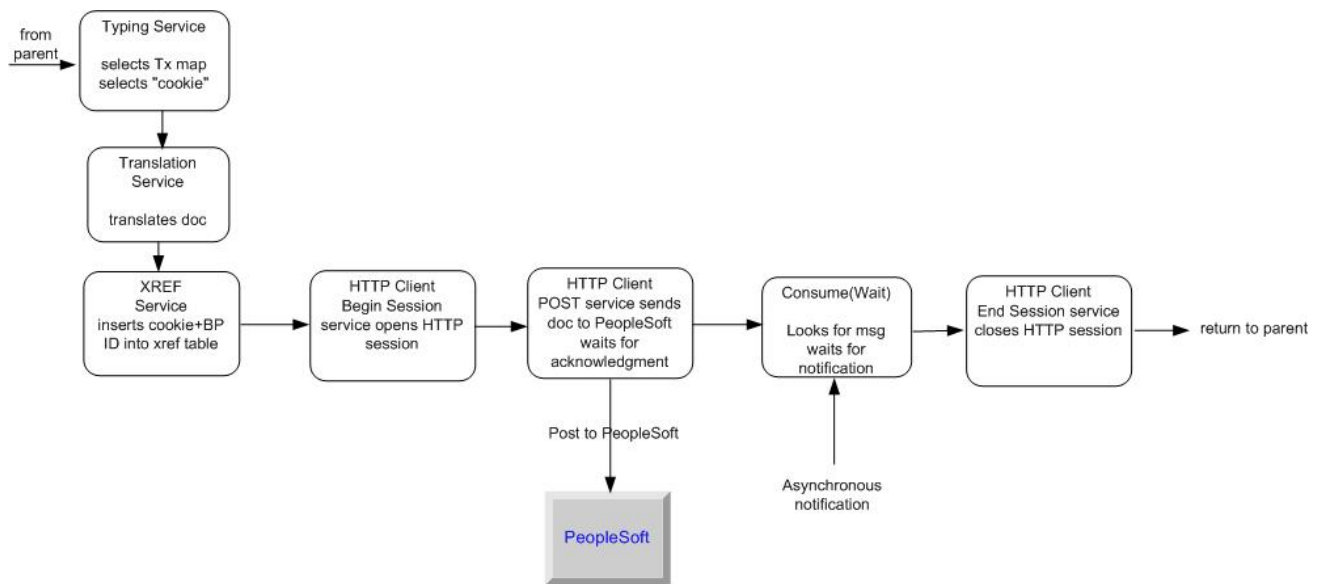
As it runs, the PS Send business process performs the following steps:

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 Sterling Integrator services to perform its work, see *Implementing the PS Send and Wait Business Process*.

PeopleSoft Receive Business Process (PS Receive)

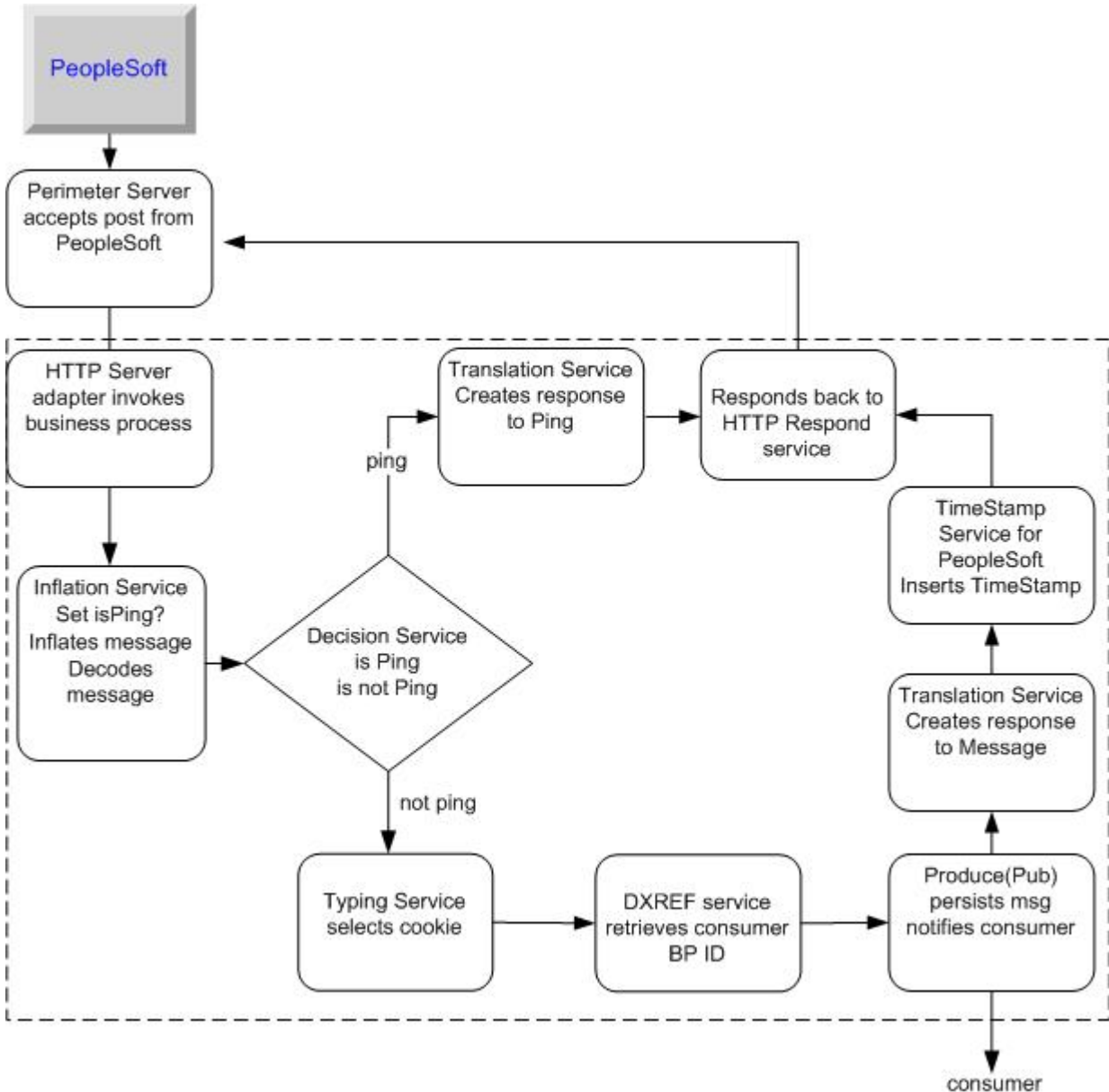
The PeopleSoft Receive (PS Receive) business process uses the HTTP Server adapter and HTTP Respond service to perform its functions. PeopleSoft never communicates directly with the PeopleSoft Receive business process. The HTTP Server adapter receives the communication and passes the information to the 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 Sterling Integrator services to perform its work, see *Implementing the PS Receive Business Process*.

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*.
 - PS Send and Wait business process. See *Implementing the PS Send and Wait Business Process*.
 - PS Receive business process. See *Implementing the PS Receive Business Process*.

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 Sterling Integrator services:

- Typing service
- Translation service
- HTTP Client Begin Session service
- HTTP Client POST service
- HTTP End Session service

Typing Service for PS Send

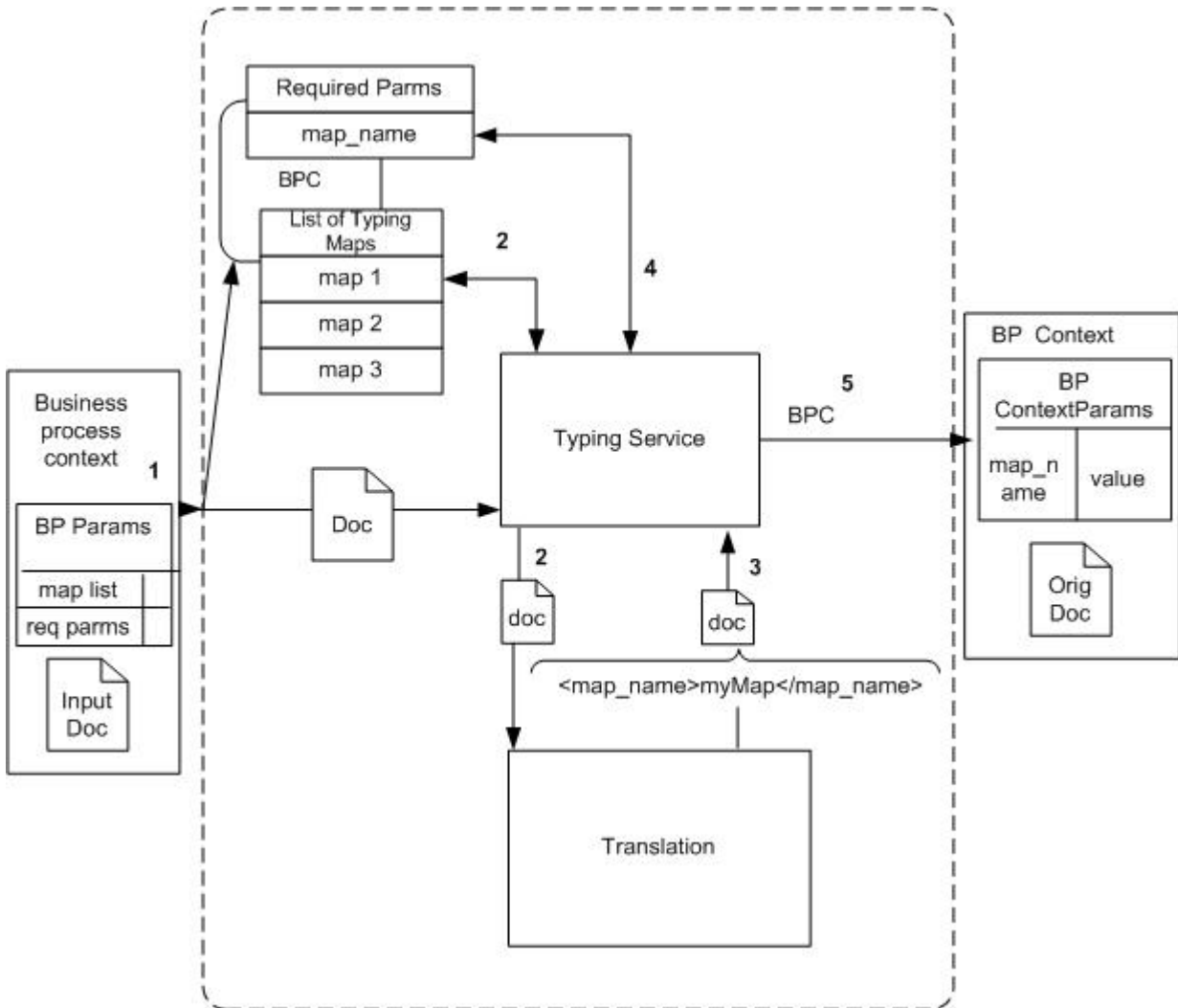
The Typing service enables Sterling Integrator to determine, at run time, which translation to perform on the input document. The service determines the input document type and sets a business process parameter (map_name) indicating which map to use to translate the document.

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.

1. The Typing service searches the output of the document translation for the `map_name` element, which is specified in the `required_parmlist` parameter.

2. 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 Sterling Integrator typing map.

Before using the PS Send business process, you must create a typing map in the Sterling Integrator Map Editor for each message supported by the business process.

The following procedure assumes that you have a working knowledge of the Sterling Integrator 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 Sterling Integrator.
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 Sterling Integrator-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 Sterling Integrator map. Before using the business process, you must create a translation map in the Sterling Integrator 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*.

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 Sterling Integrator.
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 Sterling Integrator 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 Sterling Integrator 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	#map_name = "fixed_sales_order_load";
NA	NA	Pre-session	Declares a global variable available to all rules	string[100] theBusinessUnit;
theBusinessUnit	Input (left)	Extended	Gives the global variable a value	theBusinessUnit=#BUSINESS_UNIT_2;
XREFCookie	Output(right)	Extended	Sets the XREF cookie element to the global variable	#XREFCookie =theBusinessUnit;

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 Sterling Integrator.
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"/>
</operation>
```

```

<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*.

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*.

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*.
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 Sterling Integrator 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, Sterling Integrator 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*.

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 Sterling Integrator-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 a request initiated by Sterling Integrator, 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 Sterling Integrator.

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 Sterling Integrator.
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 Sterling Integrator 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>
```

PGP Package Service

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Package service, in conjunction with the PGP Server 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 Sterling Integrator.
Requires third-party files?	No
Platform availability	All supported Sterling Integrator platforms, with the following restrictions: For NAI McAfee eBusiness Server 8.1: <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 For NAI McAfee eBusiness Server 8.5: <ul style="list-style-type: none"> • Solaris 9 or later • IBM AIX 4.2 or later • HP-UX 10.20 or later For NAI McAfee eBusiness Server 8.5.1: <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

For NAI McAfee eBusiness Server 8.6:

- Windows 2000 with Service Pack 4 or later
- Red Hat ES 3.0
- SunOS 5.9
- HP-UX 11.0
- AIX 5.1 or later

For Massachusetts Institute of Technology (MIT) Command Line Freeware 6.5.8:

- Microsoft Windows NT version 4.0 or later (Service Pack 3 or later)
- Microsoft Windows 2000
- 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

For PGP Corporation PGP® Command Line 9.5:

- Microsoft Windows 2000 (SP4)
- Microsoft Windows 2003 (SP1)
- Microsoft Windows XP (SP2)
- 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

For PGP Corporation PGP® Command Line 9.8:

- Windows 2000 (SP4)
- Windows Server 2003 (SP1)
- Windows XP (32- and 64-bit)
- Windows Vista (32- and 64-bit)
- HP-UX 11i and above (PA-RISC and Itanium)
- IBM AIX 5.2 and 5.3
- RedHat Enterprise Linux 3.0 and above (x86 only and x86_64)
- Fedora Core 3 and above (x86_64 only)
- Sun Solaris 9 (SPARC only) and Solaris 10 (SPARC, x86, and x86_64)
- Apple Mac OS X 10.4.x and 10.5.x (Universal binary)

For PGP Corporation PGP® Command Line 10.1:

- Windows 2000 SP4
- Windows Server 2003 (32- and 64-bit) SP2

	<ul style="list-style-type: none"> • Windows Server 2008 • Windows XP (32- and 64-bit) SP3 • Windows Vista (32- and 64-bit) SP2 • Windows 7 (32- and 64-bit) • HP-UX 11i and above (PA-RISC and Itanium) • IBM AIX 5.3 and 6.1 • RedHat Enterprise Linux 5.0 (x86 and x86_64) • Fedora Core 6 (x86_64 only) • Sun Solaris 9 (SPARC) and Solaris 10 (SPARC, x86, and x86_64) • SLES (SUSE Linux Enterprise Server 9 SP4 and 10 SP2 x86) • Apple Mac OS X 10.5.x and 10.6.x (Intel-based systems only)
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) • McAfee E-Business Server (version 8.5) • McAfee E-Business Server (version 8.5.1) • McAfee E-Business Server (version 8.6) • PGP® Command Line Freeware (version 6.5.8) • PGP® Command Line (version 9.5) • PGP® Command Line (version 9.8) • PGP® Command Line (version 10.1) <hr/> <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?	<p>This service does not initiate business processes. This service cannot be used without a business process.</p>
Invocation	<p>A user who has permission to perform this activity must execute the business process that invokes this service.</p>
Business process context considerations	<p>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.</p>
Returned status values	<p>Returned status values:</p> <ul style="list-style-type: none"> • 0 - Success • 1- Error <p>See <i>Advanced Status Messages</i> for a list of advanced statuses. Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrorCode].</p>
Restrictions	<p>None</p>

Persistence level	None
Testing considerations	<p>Create the profile in the PGP Server 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. A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Sterling Integrator. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client. 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 Sterling Integrator PGP Server Manager. See *PGP Server Manager*.
3. Create a PGP Package service configuration. See *Managing Services and Adapters*.
4. Configure the service. See *Configuring the PGP Package Service*.
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 cmdline2svcname 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 cmdline2svcname field is defined in the Command Line Adapter 2.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the cmdline2svcname field is defined in the Command Line Adapter 2.
profile_name	Name of PGP profile from the PGP Server 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, conv_keymap_name is used). Valid values are: IDEA, CAST5, 3DES, AES128, AES196, AES256, Twofish. Default is IDEA. Optional.
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

Field	Description
	<p>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>
pgp_partner_name	<p>The partner name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected partner profile. Required if you specify a value in the pgp_sponsor_name parameter. The values you specify in the GPM override the values you specify in the profile.</p>
pgp_sponsor_name	<p>The sponsor name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected sponsor profile. Required if you specify a value in the pgp_partner_name parameter. The values you specify in the GPM override the values you specify in the profile.</p>
tmpDir	<p>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.</p>
clearsig	<p>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.</p>
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.

Field	Description
	Optional.
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. 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. 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. To create sda files with PGP Command Line (version 9.5), set the target_platform parameter (described later in this table). The default file extension is .exe.</p> <hr/> <p>Note: The sda file cannot exceed 4 GB after compression.</p> <hr/> <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. 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. The default extension is .pga. 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 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>
        <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>
          <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>
        <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>
```

```
</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>
```



```

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

```

Business Process Example - PGP Partner and PGP Sponsor

The following business process uses the PGP Partner and PGP Sponsor services to encrypt and sign documents.

```

<process name="use_partner_sponsor">
  <operation name="PGP Package Service">
    <participant name="PGPPackageService"/>
      <output message="PGPPackageServiceTypeInputMessage">
        <assign to="pgp_partner_name">partner</assign>
        <assign to="pgp_sponsor_name">sponsor</assign>
        <assign to="profile_name">pgp</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </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

Status	Description
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 Server 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

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Unpackage service, in conjunction with the PGP Server 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 Server Manager</i> .
Preconfigured?	Yes
Requires third-party files?	No
Platform availability	All supported Sterling Integrator platforms, with the following restrictions: For NAI McAfee eBusiness Server 8.1: <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 For NAI McAfee eBusiness Server 8.5: <ul style="list-style-type: none">• Solaris 9 or later• IBM AIX 4.2 or later• HP-UX 10.20 or later For NAI McAfee eBusiness Server 8.5.1: <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

For NAI McAfee eBusiness Server 8.6:

- Windows 2000 with Service Pack 4 or later
- Red Hat ES 3.0
- SunOS 5.9
- HP-UX 11.0
- AIX 5.1 or later

For Massachusetts Institute of Technology (MIT) Command Line Freeware 6.5.8:

- Microsoft Windows NT version 4.0 or later (Service Pack 3 or later)
- Microsoft Windows 2000
- 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

For PGP Corporation PGP® Command Line 9.5:

- Microsoft Windows 2000 (SP4)
- Microsoft Windows 2003 (SP1)
- Microsoft Windows XP (SP2)
- 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

For PGP Corporation PGP® Command Line 9.8:

- Windows 2000 (SP4)
- Windows Server 2003 (SP1)
- Windows XP (32- and 64-bit)
- Windows Vista (32- and 64-bit)
- HP-UX 11i and above (PA-RISC and Itanium)
- IBM AIX 5.2 and 5.3
- RedHat Enterprise Linux 3.0 and above (x86 only and x86_64)
- Fedora Core 3 and above (x86_64 only)
- Sun Solaris 9 (SPARC only) and Solaris 10 (SPARC, x86, and x86_64)
- Apple Mac OS X 10.4.x and 10.5.x (Universal binary)

For PGP Corporation PGP® Command Line 10.1:

- Windows 2000 SP4
- Windows Server 2003 (32- and 64-bit) SP2
- Windows Server 2008

	<ul style="list-style-type: none"> • Windows XP (32- and 64-bit) SP3 • Windows Vista (32- and 64-bit) SP2 • Windows 7 (32- and 64-bit) • HP-UX 11i and above (PA-RISC and Itanium) • IBM AIX 5.3 and 6.1 • RedHat Enterprise Linux 5.0 (x86 and x86_64) • Fedora Core 6 (x86_64 only) • Sun Solaris 9 (SPARC) and Solaris 10 (SPARC, x86, and x86_64) • SLES (SUSE Linux Enterprise Server 9 SP4 and 10 SP2 x86) • Apple Mac OS X 10.5.x and 10.6.x (Intel-based systems only)
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) • McAfee E-Business Server (version 8.5) • McAfee E-Business Server (version 8.5.1) • McAfee E-Business Server (version 8.6) • PGP® Command Line Freeware (version 6.5.8) • PGP® Command Line (version 9.5) • PGP® Command Line (version 9.8) • PGP® Command Line (version 10.1) <hr/> <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?	<p>This service does not initiate business processes. This service cannot be used without a business process.</p>
Invocation	<p>A user who has permission to perform this activity must execute the business process that invokes this service.</p>
Business process context considerations	<p>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.</p>
Returned status values	<p>Returned status values:</p> <ul style="list-style-type: none"> • 0 - Success • 1- Error <p>See <i>Advanced Status Messages</i> for a list of advanced statuses. Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrorCode] .</p>
Restrictions	<p>None</p>

Persistence level	None
Testing considerations	<p>Create the profile in the PGP Server 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. A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Sterling Integrator. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client. 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: <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: <code>java -jar CLA2Client.jar <port> [debug]</code> <p>For example: <code>java -jar CLA2Client.jar 15699 debug</code></p> <hr/> <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 Server 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 cmdline2svcname 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 cmdline2svcname 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.
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.
pgp_partner_name	The partner name used in encryption and signing. If specified, the business processes the parameters you specify in the selected partner profile. Required if you specify a value in the pgp_sponsor_name parameter. The values you specify in the GPM override the values you specify in the profile.

Field	Description
pgp_sponsor_name	The sponsor name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected sponsor profile. Required if you specify a value in the pgp_partner_name parameter. The values you specify in the GPM override the values you specify in the profile.
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.
DocumentPGP/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.
ErrorCodePGP/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>
        <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>
```

Business Process Example - PGP Partner and PGP Sponsor

The following business process uses the PGP Partner and PGP Sponsor services to decrypt and verify documents.

```
<process name="use_partner_sponsor">
  <operation name="PGP Unpackage Service">
    <participant name="PGPUnpackageService" />
    <output message="PGPUnpackageServiceTypeInputMessage">
      <assign to="pgp_partner_name">partner</assign>
      <assign to="pgp_sponsor_name">sponsor</assign>
      <assign to="profile_name">pgp</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</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
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.
160	Complete failure during a file wipe.

Status	Description
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 Server 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.

Adapter for PeopleSoft CRM CIC

The Adapter for PeopleSoft CRM CIC (formerly the Vantive adapter) enables a Sterling Integrator 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 Sterling Integrator 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	Available for: <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> .
Business process context considerations	None
Returned status values	Basic status: Success or Failure Advanced Statuses: see <i>Error Messages</i> .

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.
 - Sterling Integrator 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 Sterling Integrator. 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 Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator.
5. Sterling Integrator 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:

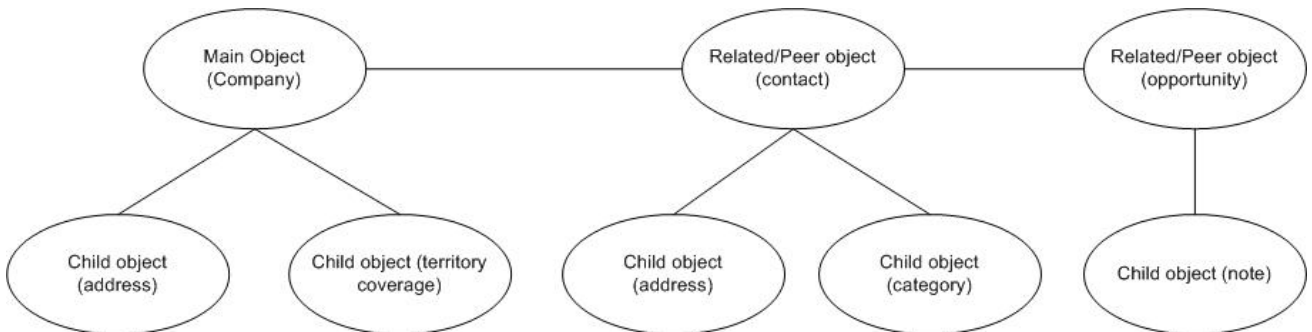
- Object hierarchy
- Object type
- Record set or record
- Field

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.



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_CRMAdapter>
  <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_CRMAdapter>
```

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_CRMAdapter>
  <results status="success" statusText="2 Record(s) Created."/>
</PS_CRMAdapter>
```

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_CRMAdapter>
  <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_CRMAdapter>
```

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_CRMAdapter>
  <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_CRMAdapter>
```

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_CRMAdapter>
  <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_CRMAdapter>
```

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_CRMAdapter>
  <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_CRMAdapter>
```

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*. The following XML example shows an output DTD for reading fields on a child object, customer address. The result returns one to many records, each containing two fields, Street 1 and Street 2.

```
<?xml version='1.0' encoding='UTF-8' ?>
<!ELEMENT PS_CRMAdapter (results)>
<!ELEMENT results (result*)>
<!ATTLIST results status CDATA #IMPLIED
                statusText CDATA #IMPLIED >
<!ELEMENT result (field.1, field.2)>
<!ELEMENT field.1 (#PCDATA)>
<!ATTLIST field.1 name CDATA #FIXED "Street1">
<!ELEMENT field.2 (#PCDATA)>
<!ATTLIST field.2 name CDATA #FIXED "Street2">
```

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	Connection to the PeopleSoft CRM CIC server was not completed for one of the following reasons: <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*.
3. Create an Adapter for PeopleSoft CRM CIC configuration. See *Creating a Service Configuration*.

- Configure the Adapter for PeopleSoft CRM CIC. See *Configuring the Adapter for PeopleSoft CRM CIC for Application*.

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.l`
- 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 Sterling Integrator installed:

```
./install3rdParty.sh <Vendor> <Version> -j <location of
the .jar>
```

The following figure shows an example of the script for the PeopleSoft CRM CIC `.jar` files:

```
./install3rdParty.sh PeopleSoftCRM 9.0 -j /home/PeopleSoftCRM/vanjavi.jar
```

For the PeopleSoft CRM CIC shared libraries, run:

```
./install3rdParty.sh <Vendor> <Version> -l <location of
the lib>
```

The following figure shows an example of the script for the PeopleSoft CRM CIC shared libraries:

```
./install3rdParty.sh PeopleSoftCRM 9.0 -l /home/PeopleSoftCRM/solaris
/vanjavi.so
```

Configuring the Adapter for PeopleSoft CRM CIC for Sterling Integrator

The following table describes the fields used to configure the Adapter for PeopleSoft CRM CIC in Sterling Integrator:

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.

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.
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.
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:

```
<operation name="PeopleSoft CRM ">
  <participant name="PSCRMTestAdapter"/>
  <output message="outputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inputMessage">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

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:

```
<process name="PSCRMAdapter_Joe">
  <sequence>
    <operation name=" PeopleSoft CRM ">
      <participant name=" PSCRMTestAdapter "/>
      <output message="outputMessage">
        <assign to="Login">Joe</assign>
        <assign to="Password">12345</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inputMessage">
        <assign to="." from="*"></assign>
      </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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none">• 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	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. 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.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Sterling Integrator 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 Sterling Integrator, 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 <http://www.sterlingcommerce.com> and follow the links to Customer Support.

The following example shows the PurgeAll business process as delivered with Sterling Integrator. Sterling Integrator 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>
    </operation>
  </sequence>
</process>
```

```

    <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>
    <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>
#: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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • Success – If release is successful • Failure – If release is unsuccessful

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* 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. 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> 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>
    <operation name="release">
      <participant name="ReleaseService"/>
      <output message="outmsg">
        <assign to="TARGET">TempVariables</assign>
      </output>
      <input message="inmsg"/>
    </operation>
  </sequence>
</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.bp



Service Editor - release

Name: release

Config: ReleaseService

Message To Service: Message From Service

Output Msg: Messages Only

Message Name: outmsg

Advanced Editor : release

Name	Value	Use XPath?
TARGET	TempVariables	<input type="checkbox"/>

Click **Advanced** to open the Advanced Editor.

Specify the TARGET parameter and its corresponding value.

Advanced

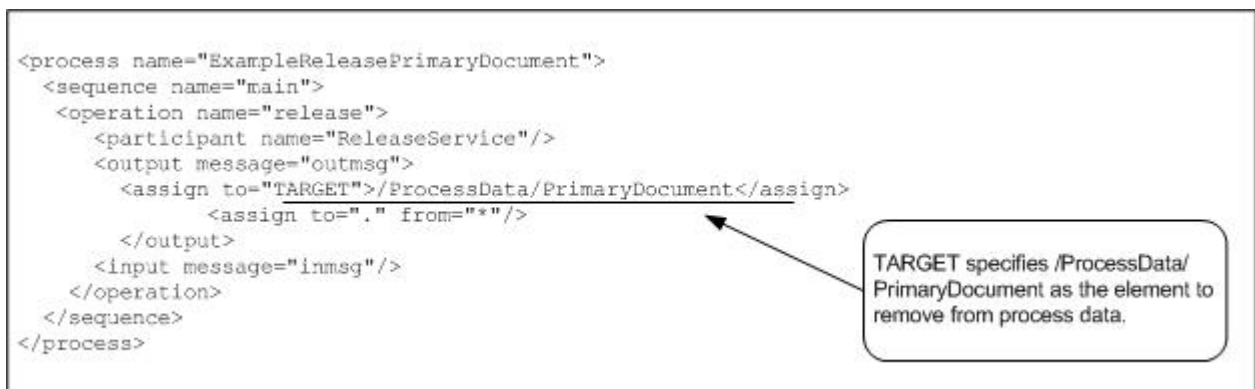
Removing a Primary Document Reference from Process Data

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.



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:

ExampleReleasePrimaryDocument.bp

Service Editor-release

Name: release

Config: ReleaseService

Message To Service: Message From Service

Output Msg: Obtain Message first, then Proce

Message Name: outmsg

Advanced Editor : release

Name	Value	Use XPath?
TARGET	/ProcessDataPrimaryDocument	<input type="checkbox"/>

Specify the TARGET parameter and its corresponding value.

Click **Advanced** to display the Advanced Editor.

Advanced

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* for the input document.

The following example illustrates the business process using BPML:

```

<process name="ExampleReleaseTemporaryDocument">
  <sequence name="main">
    <assign to="TempDocumentHolder" from="DocToDOM(PrimaryDocument)" append="true"/>
    <operation name="release">
      <participant name="ReleaseService"/>
      <output message="outmsg">
        <assign to="TARGET">TempDocumentHolder</assign>
      </output>
      <input message="inmsg"/>
    </operation>
  </sequence>
</process>

```

The **assign** statement loads the primary document into process data under the element TempDocumentHolder.

TARGET specifies TempDocumentHolder as the element to remove from process data.

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>

```

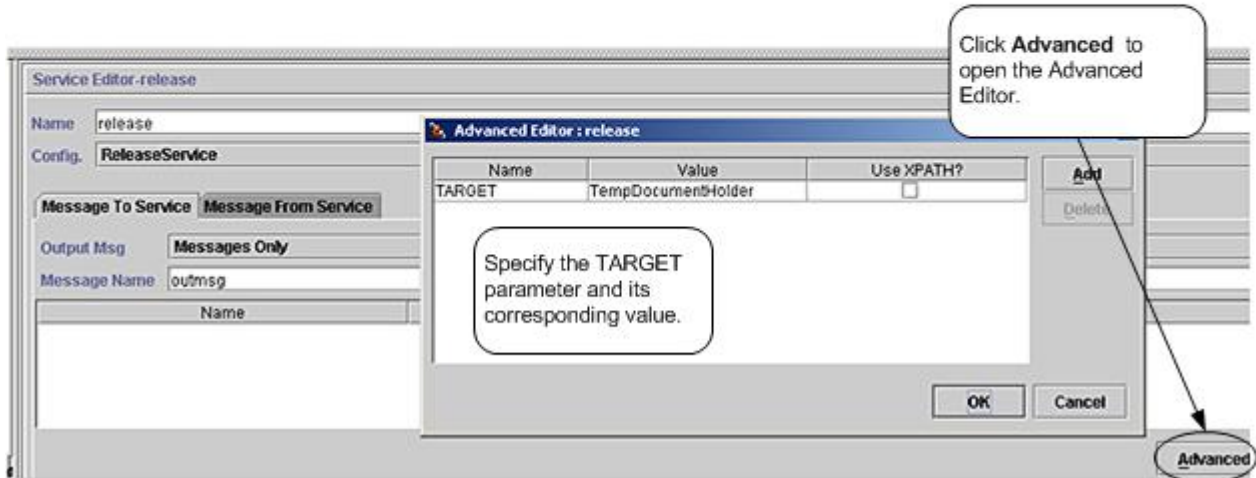
After the Release service is processed, the process data looks like the following, with references to the original primary document and the temporary document removed:

```

<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>
```

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, <code>RPT_<ReportConfigName>.<Format></code> . For example, if the format is <code>PDF</code> and the Report Configuration name is <code>TestConfig</code> , 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 Sterling Integrator 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	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. <hr/> 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*.
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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator platforms Note: You must restore data that was backed up using the same version and patch level as your current installation of Sterling Integrator. 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator, 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	<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>

Field	Description
Restore Thread Pool Size	Maximum number of threads the Restore Business Process service will use for its processing. Required. Default is 8.
Run as User	<p data-bbox="850 333 1477 474">Applies to the scheduling of the business process. Type the user ID to associate with the schedule, or click the  icon and select a user ID from the list. Valid value is any valid Sterling Integrator user ID.</p> <hr/> <p data-bbox="850 506 1477 657">Note: This parameter allows someone who does not have rights to a specific business process to run it. If you select Admin 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 data-bbox="850 779 1477 835">Information about scheduling the business process invoked by the Restore Business Process service. Valid values:</p> <ul data-bbox="850 852 1477 1690" style="list-style-type: none"> <li data-bbox="850 852 1477 909">• Do not use schedule If this field is selected, the service does not run on a schedule. <li data-bbox="850 926 1477 1104">• 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. <li data-bbox="850 1121 1477 1299">• 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. <li data-bbox="850 1316 1477 1495">• 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. <li data-bbox="850 1512 1477 1690">• 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.

Remote Method Invocation (RMI) Adapter

Note: In the Sterling Integrator 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 RMIIOAdapter. 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	Receives requests from remote RMI clients to start business processes. You can start a business process within Sterling Integrator 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. The RMI client can use the RMI adapter over the Java Remote Method Protocol (JRMP).
Preconfigured?	No
Requires third party files?	No
Platform availability	All supported Sterling Integrator platforms
Related services	No
Application requirements	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. 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.
Initiates business processes?	Provides the remote RMI Client with the following two options for starting a business process: <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	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.
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 Sterling Integrator 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 Sterling Integrator. There is no automatic secure channel associated with RMI.

If Sterling Integrator 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*.
- 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.

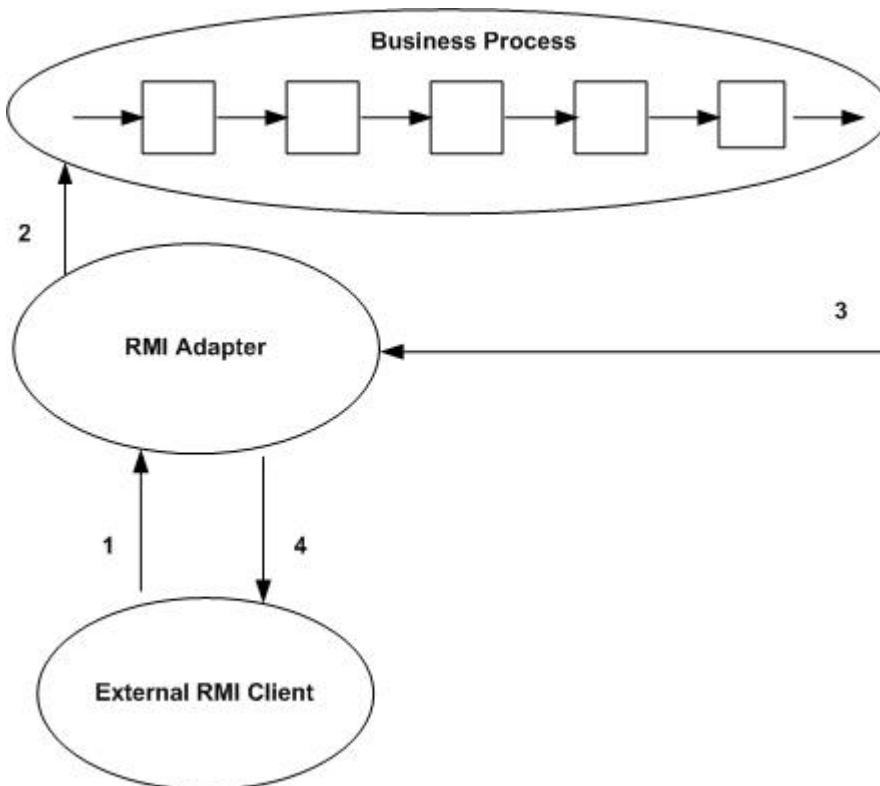
Synchronous Requests

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*.
3. Specify RMI method parameters. For information, see *Starting a Business Process*.
4. Set system properties to make connections to the RMI adapter. For information, see *Making Connections to the RMI Adapter*.
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*.
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 `RMIIOAdapterDocument` object; otherwise, null is returned.

```

RMIIOAdapterDocument 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.

SAP XI Adapter

The SAP XI adapter, in conjunction with the GIS Resource Adapter (GIS RA), provides integration capability between Sterling Integrator 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 Sterling Integrator to the SAP XI component of SAP NetWeaver. The SAP XI adapter communicates with SAP XI through the GIS RA (Resource Adapter) supplied with Sterling Integrator 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 Sterling Integrator and SAP XI. This allows a tight integration between Sterling Integrator and NetWeaver-based SAP solutions.
Usage example	Sterling Integrator 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 Sterling Integrator and SAP XI. Both adapters require configuration steps to be performed after installation.
Requires third party files?	<p>The SAP XI adapter requires three library files from your SAP XI installation:</p> <ul style="list-style-type: none"> • sapj2eeclient.jar • exception.jar • logging.jar <p>See <i>Installing the SAP XI Library Files</i> for more information.</p> <hr/> <p>Note: The SAP XI adapter does not require the SAP JCo libraries.</p>

Platform availability	All supported Sterling Integrator 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 <code>SAP_J2EE_ADMIN</code> to the SAP XI user account. Similarly, a Sterling Integrator account (User ID and password) is required for the GIS RA to communicate with Sterling Integrator. The SAP XI adapter must be registered in the Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator</p> <p>SAP XI triggers an outbound (out of SAP) message. The SAP RA adapter sends the message to Sterling Integrator. 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>Sterling Integrator to SAP XI</p> <p>A Sterling Integrator 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>Returned status values:</p> <ul style="list-style-type: none"> • Success – SAP XI ended successfully • Error – Various advanced status values plus process data content <p>Advanced Status: The SAP XI adapter returns one of the following messages if an error occurs during message processing:</p> <ul style="list-style-type: none"> • <code>GA_INITIALIZATION_FAILED</code> – Error during SAP XI adapter, including communication failures such as “bind failed”. • <code>GA_SYNC_SEND_FAILED</code> – Error using the <code>sendSync</code> method, including communication failures.

	<ul style="list-style-type: none"> • 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. GIS RA 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 Sterling Integrator 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 Sterling Integrator and SAP NetWeaver is actually accomplished by two adapters:

- SAP XI adapter – A stateful adapter which is deployed and runs in Sterling Integrator.
- 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 Sterling Integrator to SAP XI and receive synchronous responses from SAP XI. (Configured as an inbound adapter.)
- Receive messages from SAP XI, initiate business processes in Sterling Integrator, 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 Sterling Integrator SAP XI adapter and receive synchronous responses.
- Receive messages from the Sterling Integrator 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 Sterling Integrator. The connectivity depends on whether you are sending or receiving messages:

1. An inbound SAP XI adapter configuration in Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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
</attachments>
```

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:

```
<attachments>
  attachment_1
</attachments>
```

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:

```
<attachment_n>
  <document>
    PrimaryDocument
  </document>
</attachment_n>
```

The structure for a document node with optional index attributes is:

```
<attachment_n>
  <document index=n>
    PrimaryDocument
  </document>
  <properties index=n>
```

```
    BodyProperties
  </properties>
</attachment_n>
```

Body properties describe details of a payload message or attachment message. The structure for body properties is:

```
<BodyName>name</BodyName>
<BodyDescription>description</BodyDescription>
<BodyVersion>version</BodyVersion>
<BodySchema>schema</BodySchema>
<BodyEncoding>encoding</BodyEncoding>
<BodyContentType>type</BodyContentType>
<IsXMLContent>true or false</IsXMLContent>
```

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
<SAPXIResponse>
  HeaderDetails
  <properties>
    BodyProperties
  </properties>
  Attachments (optional)
</SAPXIResponse>
```

Header Details

```
<Header>
  <ToService>detail</ToService>
  <FromService>detail</FromService>
  <ToParty>detail</ToParty>
  <FromParty>detail</FromParty>
  <MessageID>detail</MessageID>
  <Action>detail</Action>
  <ActionNS>detail</ActionNS>
</Header>
```

Outbound Messages

Messages sent from SAP XI to Sterling Integrator 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:

```
PrimaryDocument
SAP XI Outbound Structure
<OriginatorID>id</OriginatorID>
```

The SAP XI Outbound structure is:

```
<SAPXIOutbound>
  Header
  <properties>
    BodyProperties
  </properties>
  Attachments (Optional)
</SAPXIOutbound>
```

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*.
2. **XICallType** must be set to **SYNCHRONOUS_RESPONSE**. See *Setting Up the SAP XI Adapter in the GPM*.

The Response message consists of the contents of the primary document, and optionally, attachments.

See *Business Process Examples* for more information.

Implementing the SAP XI Adapter

To implement the SAP XI adapter, complete the following tasks:

1. Install the third party files. See *Installing the SAP XI Library Files*.
2. Create a configuration of the SAP XI adapter in Sterling Integrator. See *Managing Services and Adapters*. For information about the fields specific to this adapter, see *Configuring the SAP XI Adapter*.
3. Specify field settings for the adapter configuration in the Sterling Integrator Admin Console and in the GPM as necessary.
4. Implement the SAP RA adapter in SAP XI. See *Installing and Configuring the GIS Resource Adapter within SAP XI*.

Installing the SAP XI Library Files

The SAP XI adapter requires three third-party libraries. Obtain the necessary .jar files from your SAP XI installation and install them on your Sterling Integrator system.

Perform the following steps:

1. On your SAP XI system, change to one of the following directories:
 - /usr/sap/systemNumber/DVEBMGS00 /j2ee/j2eeclient/signed (recommended)
 - /usr/sap/systemNumber/DVEBMGS00/j2ee/j2eeclient
2. Copy the following files to your Sterling Integrator system:
 - sapj2eeclient.jar
 - exception.jar
 - logging.jar

Note the location where you save the files.

3. Stop Sterling Integrator.

- Install the libraries by running commands similar to the following from the Sterling Integrator installation directory:

UNIX or Linux

- `cd install_dir/bin`
- `./install3rdParty.sh sapxi 30 -j path/sapj2eeclient.jar`
- `./install3rdParty.sh sapxi 30 -j path/exception.jar`
- `./install3rdParty.sh sapxi 30 -j path/logging.jar`

Windows

- `cd install_dir\bin`
- `install3rdParty.cmd sapxi 30 -j path\sapj2eeclient.jar`
- `install3rdParty.cmd sapxi 30 -j path\exception.jar`
- `install3rdParty.cmd sapxi 30 -j path\logging.jar`

- Start Sterling Integrator.

Configuring the SAP XI Adapter

You must specify field settings in Sterling Integrator, 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 Sterling Integrator. 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	Select one of the following 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. <hr/> <p>Note: See <i>Managing Services and Adapters</i>.</p>
Service Type (ServiceType)	Identifies the SAP XI adapter as an inbound (to SAP XI) or outbound (from SAP XI) adapter. Required. Valid values are: <ul style="list-style-type: none"> • SAP XI Inbound – (Default) Message flow from Sterling Integrator to SAP XI.

Field	Description
	<ul style="list-style-type: none"> • SAP XI Outbound – Message flow from SAP XI to Sterling Integrator.
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 Sterling Integrator.
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.
SAP XI JNDI Host Name and Port (Host:Port) (JNDIRemoteAddress)	Host and JNDI port of the remote SAP XI JNDI Server. Required. Syntax is Host : Port
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 com.sterlingcommerce.woodstock.services.sapxira.
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 com.sap.engine.services.jndi.InitialContextFactoryImpl.
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 SAP_J2EE_ADMIN). 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 Host : Port.
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 com.sterlingcommerce.woodstock.services.sapxiga.
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 org.jnp.interface.NamingContextFactory. Change default to the following value: com.sap.engine.services.jndi.InitialContextFactoryImpl.

Field	Description
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 Sterling Integrator 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.

Field	Description
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 Sterling Integrator 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*
- *Deploying the SDA in SAP NetWeaver*
- *Adding the RA to the System Landscape Directory*
- *Creating a Repository Namespace*
- *Creating an Adapter Object*

Copying the SDA to SAP NetWeaver

To copy the SDA to SAP NetWeaver:

1. Locate the following SDA file on the machine where Sterling Integrator 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

To deploy 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 Sterling Integrator.
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)	Sterling Integrator JNDI Host Name and Port (Host:Port). Required.
GIS JNDI Key	Sterling Integrator JNDI Key. Required. Default is com.sterlingcommerce.woodstock.services.sapxiga
GIS JNDI Context	Sterling Integrator JNDI Context. Required. Default is org.jnp.interfaces.NamingContextFactoryChange default to the following value: com.sap.engine.services.jndi.InitialContextFactoryImpl
GIS JNDI User Name	Sterling Integrator JNDI User Name. Required.
GIS JNDI Password	Sterling Integrator 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

Field	Description
	• 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 (Sterling Integrator 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="serverNode:169ca65:109f0bef19c:2f25" />
  </doc1>
  <att2>
    <message>This is my attachment 2 message</message>
  </att2>
  <doc2>
    <PrimaryDocument SCIObjectID="serverNode:169ca65:109f0bef19c:2f47" />
  </doc2>
  <attachments>
    <attachment_1>
      <document index="1">
        <PrimaryDocument SCIObjectID="serverNode:169ca65:109f0bef19c:2f25" />
      </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>
    <attachment_2>
      <document index="2">
        <PrimaryDocument SCIObjectID="serverNode:169ca65:109f0bef19c:2f47" />
      </document>
    </attachment_2>
  </attachments>
</ProcessData>
```

```

</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>serverNode:50004</JNDILocalAddress>
<RemoteJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl
</RemoteJNDIContext>
<ServiceType>Inbound</ServiceType>
<LocalJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl
</LocalJNDIContext>
<JNDIRemoteAddress>serverNode:50004</JNDIRemoteAddress>
<LocalRegistryType>EXTERNAL_REG</LocalRegistryType>
<RemoteUser>userName</RemoteUser>
<LocalUser>userName</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>
</process>
```



```

<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 Sterling Integrator):

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="serverNode: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=
          "serverNode: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=
          "serverNode: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>serverNode.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="serverNode:169ca65:109de7de0fd:-26e9"/>
</doc1>
<attachments>
  <attachment_1>
    <document index="1">
      <PrimaryDocument SCIObjectID="serverNode: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:

```

<SAPXIOOut>
  <SyncWaitTimeout>60</SyncWaitTimeout>
  <BootstrapBP>sapxiout_response</BootstrapBP>
  <LocalPassword>*****</LocalPassword>
  <RemotePassword>*****</RemotePassword>
  <JNDILocalAddress>serverNode:50004</JNDILocalAddress>
  <RemoteJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl
</RemoteJNDIContext>
  <ServiceType>Outbound</ServiceType>
  <LocalJNDIContext>com.sap.engine.services.jndi.InitialContextFactoryImpl
</LocalJNDIContext>
  <JNDIRemoteAddress>serverNode:50004</JNDIRemoteAddress>
  <LocalRegistryType>EXTERNAL_REG</LocalRegistryType>
  <RemoteUser>userName</RemoteUser>
  <SyncWait>Yes</SyncWait>
  <LocalUser>userName</LocalUser>
  <JNDILocalKey>com.sterlingcommerce.woodstock.services.sapxigareturn
</JNDILocalKey>
  <XMLVersion>1.0</XMLVersion>
  <JNDIRemoteKey>com.sterlingcommerce.woodstock.services.sapxirareturn
</JNDIRemoteKey>
  <XICallType>SYNCHRONOUS_RESPONSE</XICallType>
  <OriginatorID>serverNode.169ca65.109de7de0fd.F5647</OriginatorID>
  <AttachmentRoot>attachments</AttachmentRoot>
  <FromParty/>
  <FromService>I03_OutboundToGIS</FromService>
  <ToParty/>
  <ToService>I03_InboundFromGIS</ToService>
  <Action>CarstensMessageInterfaceSync</Action>
  <ActionNS>http://stercomm.com/adapter/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 values: <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

Parameter	Description
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.
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 Sterling Integrator 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.

SAP Suite Adapter

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 Sterling Integrator 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	Usage includes: <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 Sterling Integrator platforms
Related services	None
Application requirements	SAP connection details (see <i>Before You Begin</i>) are required for the SAP Suite adapter to communicate with SAP.
Initiates business processes?	The SAP Suite adapter initiates a business process in Sterling Integrator 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 Sterling Integrator, 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	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> .
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 Sterling Integrator, 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, Sterling Integrator 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, Sterling Integrator provides the following business processes:

- SAP Inbound IDoc
- SAP Outbound IDoc
- SAP Inbound Delivery

For ALE/IDoc processing, Sterling Integrator 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. Sterling Integrator 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*.

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*.
- 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*.
- Create a business process that includes the EDI Develope service for deenvolving the EDI document.
- Configure an instance of the SAP Suite adapter that uses file-based IDoc RFC (see *Example SAP Suite Adapter Configuration*).
- Update the predefined SAPInbDelivery business process with the appropriate values for sending the IDoc to the SAP system. See *SAP Inbound Delivery (SAPInbDelivery.bp)* for the required updates.
- Create an FTP login authentication file required by the FTP server and place the file in the directory where Sterling Integrator is installed. See *Before You Begin*.

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:

▶ Example GPMSAPSuiteFileBasedIDOCRFC

Service Settings

Service Type	SAP Suite Adapter
Description	Example to include in the documentation
System Name	Example GPMSAPSuiteFileBasedIDOCRFC
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

Specifies the integration mode for interacting with the SAP system.

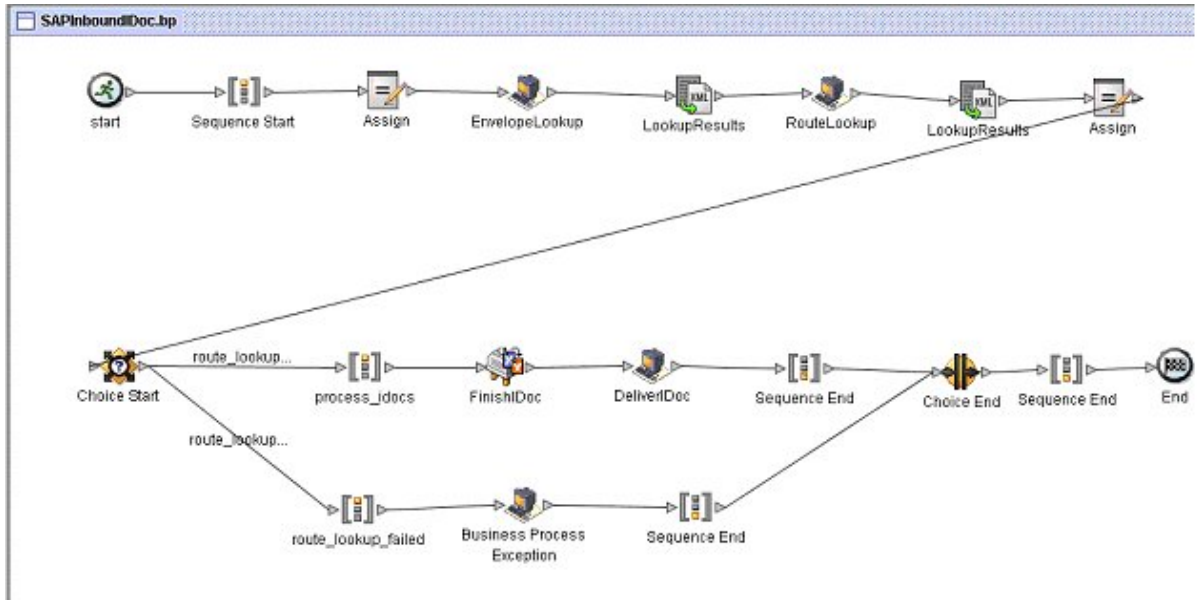
(Screen 1 of 2)

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)

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/@SCIOObjectID"/>
    <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>
</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.

```

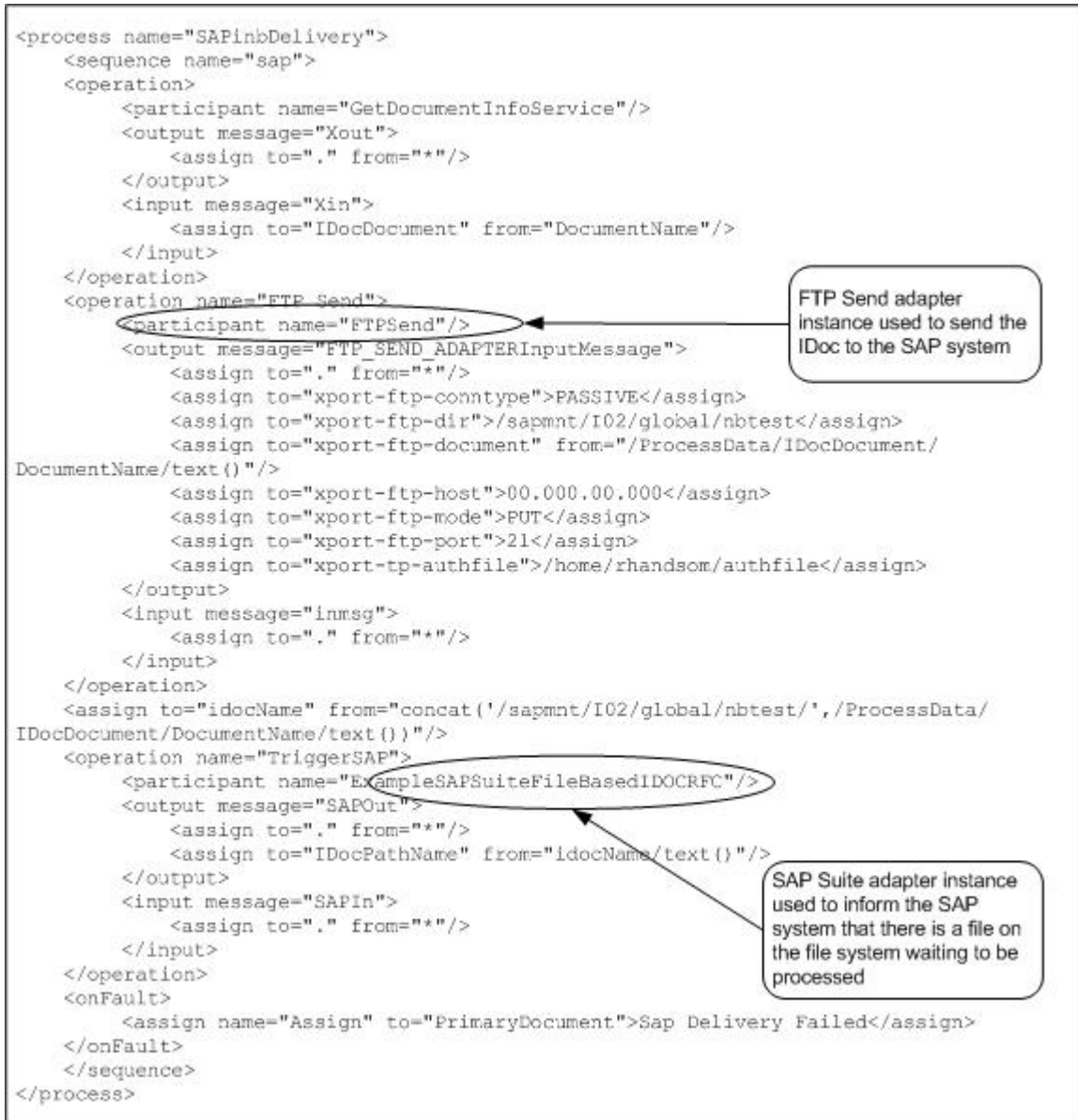
<assign name="Assign" to="PrimaryDocument" from="OrigPrimaryDoc/@SCIOBJECTID"/>
<choice name="Choice Start">
  <select>
    <case ref="route_lookup_results" activity="process_idocs"/>
    <case ref="route_lookup_results" negative="true"
activity="route_lookup_failed"/>
  </select>
  <sequence name="process_idocs">
    <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">ASYN</assign>
        <assign to="KFD_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>EDIPORT</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* 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*.
3. Create all of the maps that will be used to translate the IDocs that you are enveloping, and check them into Sterling Integrator.
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*.
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 Sterling Integrator. After implementing the predefined business processes, you must check in your business process back to Sterling Integrator.

For more information about implementing SAP business processes, see *Implementing the SAP R/3 Business Processes*.

7. Create inbound and outbound routes that describe the key fields in the IDoc that enable Sterling Integrator to route the IDoc to the appropriate trading partner. For information, see *Configuring an SAP R/3 Cross-Reference*.
8. Create inbound and outbound cross-references to the routes that enable Sterling Integrator 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*.

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 Sterling Integrator. After obtaining the SAP JCo, you must make the files available to the host system where Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator:

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	Includes: <ul style="list-style-type: none"> • librfccm.o • libsapjcorfc.so • sapjco.jar
HP-UX11	Includes: <ul style="list-style-type: none"> • librfccm.sl • libsapjcorfc.sl • sapjco.jar <p>Important: For JCo most recent 2.x.x version 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. 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	Includes: <ul style="list-style-type: none"> • librfccm.so • libsapjcorfc.so • sapjco.jar
Sun Solaris	Includes: <ul style="list-style-type: none"> • librfccm.so • libsapjcorfc.so • sapjco.jar
Microsoft Windows	Includes: <ul style="list-style-type: none"> • librfc32.dll • sapjco.jar

Platform	SAP JAVA Connector Files
	<ul style="list-style-type: none"> • sapjcorfc.dll <p>Important: For JComost recent 2.x.x version 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	<p>Includes:</p> <ul style="list-style-type: none"> • sapjco.jar • librfccm.o • libsapjcorfc.so

Installing on UNIX

To install the SAP Java Connector on UNIX:

1. Complete the procedure *Downloading the SAP Java Connector*.
2. Shut down Sterling Integrator 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 (the most recent 2.x.x version) -j directory/*.jar
```

- b. For the native libraries, type:

```
install3rdParty.sh sapjco 2_x_x (the 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*.
- If you are using the MySQL database, ensure that the database is stopped after running `install3rdParty.sh`.

4. Restart Sterling Integrator.

Installing on Windows

To install the SAP Java Connector on Windows:

1. Complete the procedure described in *Downloading the SAP Java Connector*.

2. Shut down Sterling Integrator 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 (the most recent 2.x.x version) -j directory\*.jar
```

- b. For the native libraries, type:

```
install3rdParty.cmd sapjco 2_x_x (the 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 Sterling Integrator.

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*.
2. From an iSeries command line, type `QSH` and press **Return** to get into Qshell.
3. Change to the bin directory of your Sterling Integrator instance using the following command:

```
cd /<install_dir>/bin
```

4. Type the following commands, where `<x>` is the SAP version and `<install_path>` is the directory of the file:

```
./Install3rdParty.sh sapjco 2_1_<x> -j /<install_path> /sapjco.jar -nodeploy
./Install3rdParty.sh sapjco 2_1_<x> -l /<install_path> /librfccm.o -nodeploy
./Install3rdParty.sh sapjco 2_1_<x> -l /<install_path> /libsapjcorfc.so
-nodeploy
```

5. Upon completion, type the following command:

```
./deployer.sh -x/<install_dir>/properties/deployer.xml
```

6. Verify that the `passphrase=password` has not been removed from the Sterling Integrator `install_dir/properties/security.properties` file. If it has, then add it to the file.
7. Restart Sterling Integrator.

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* 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 Sterling Integrator 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.

Field	Description
	<ul style="list-style-type: none"> • 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>For more information on Service Groups, see <i>Managing Services and Adapters</i>.</p>
SAP Integration Mode (Mode)	<p>Integration mode that enables Sterling Integrator to communicate with an SAP R/3 system and with transfer documents. Required. Valid values:</p> <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)	<p>Whether Sterling Integrator is connecting to a single SAP R/3 system or cluster of SAP R/3 systems. Required. Valid values:</p> <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
- Synchronous RFC
- File-based IDoc RFC
- ALE-based IDoc RFC
- User Properties
- RFC Server Configuration (Outbound)
- Connection Pool Settings
- Application User Settings

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	

Field	Description
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.
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 Sterling Integrator 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 Sterling Integrator 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.

Field	Description
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.
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 boxRetry 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.
Language (Lang)	Language used by the SAP R/3 account. Required. Valid value is EN (English).

Field	Description
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 Sterling Integrator 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 Sterling Integrator 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.
Register Remote Function	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 Sterling Integrator. Options are: <ul style="list-style-type: none"> • Select RFCs online from RFC list Sterling Integrator 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.

Field	Description
CheckboxSAP R/3 is a UNICODE system	To be able to interface with Unicode SAP systems, you must enable the check box. 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.

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.
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.
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 Sterling Integrator.

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 Sterling Integrator:
 - 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator is installed and where it extracts (writes)

Service/Adapter Configuration Instance Name	Parameter	Description
		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 Sterling Integrator 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 Sterling Integrator 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> .

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> .

You are now ready to check in the business processes to Sterling Integrator.

Configuring an SAP R/3 Route

Sterling Integrator 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 Sterling Integrator and an SAP EDI subsystem. The EDI subsystem triggers the SAP Suite adapter when sending IDocs to Sterling Integrator. Sterling Integrator 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 Sterling Integrator.

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	Name of the port that enables communication between Sterling Integrator and an SAP EDI subsystem. The EDI subsystem triggers the SAP Suite adapter when sending IDocs to Sterling Integrator. Sterling Integrator triggers the EDI subsystem when sending IDocs to SAP. Required. 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).
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. 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).
SAP Partner Keys	The key fields from the IDoc that Sterling Integrator uses to identify the IDoc. Required. Select all the key fields that apply. Valid values: <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 Sterling Integrator.

Configuring an SAP R/3 Cross-Reference

To enable Sterling Integrator 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 Sterling Integrator 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. <hr/> 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 Sterling Integrator.

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. <hr/> 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 Sterling Integrator.

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.

Sterling Integrator 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 Sterling Integrator 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 IDoc type or schema (or its extension) in the form of a data definition format (.ddf) or schema (.xsd) file (see *DDF in create mode* or *XSD in create mode*). 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 Sterling Integrator 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
- Sterling Integrator 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:

```
bin\runSAPClass.cmd schemaGenerator.CmdlineSAPSuiteBuilder <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 Sterling Integrator environment (that is, the path begins with `com.sterlingcommerce.Woodstock.services.sapsuite`). Therefore, the script expects, as a first parameter, the name of the Java class 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:

```
CmdlineSAPSuiteBuilder (Version 1.3)
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 (BasicType 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 alphanum. 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_Availability.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	
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

Field Name	Default Value	Non LB, M or O/LB, M or O	Notes/Used Datatype
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)

Parameter	Data Type/ HTML Type/ Validator/ Size/MaxSize	Value/ Default	Description
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>.
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

Parameter	Data Type/ HTML Type/ Validator/ Size/MaxSize	Value/ Default	Description
			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.
ServerSessionSequenceNumber	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			

Parameter	Type	Data Type/ Html Type/ Validator/ Size/ MaxSize	Value/ Default	sRFC	BAPI	RFC IDoc	ALE IDoc	Client or Server
RFCURLName	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	Single Number	0,1	O	O			S
ServerSessionID	WP	String/text		O	O			S
ServerResponse	WP	String/text		O	O			S

Export Parameters

The following table describes the business process parameters when exporting parameters:

Parameter	Data Type/ HTMLType/Validator	sRFC	BAPI	RFC IDoc	ALE IDoc
SessionID	String	O	O	-	-
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 Sterling Integrator 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

Row	Description
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.

Status	Description
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.
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.
RepositoryNotInit	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.

Status	Description
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.
TidMaintenaceError	Error during maintenance of SAP_TID table.
OpenConnectionError	Error while establishing connection.
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 Sterling Integrator 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 Sterling Integrator, it provides stability to instance server and increases the critical IDoc size limit. For example, it does not affect the stability of Sterling Integrator, 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 Sterling Integrator.

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_name>.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 Sterling Integrator 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 Sterling Integrator.

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>\bin\InstallSAPControllerWindowsService.cmd` 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 Sterling Integrator 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 Sterling Integrator is installed.
- If Sterling Integrator uses a 64-bit JVM, follow the steps described in *Modifying SAP Controller JVM Settings for 64-Bit Platforms* before starting SAP Controller (add `-d64` JVM option).
- Modify the `security.properties` file as described in *Modifying Security Properties* to make sure that Sterling Integrator 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 Sterling Integrator scripts `run.sh` (UNIX) or `startWindowsService.cmd` (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 Sterling Integrator scripts `run.sh` (UNIX) and `startWindowsService.cmd` (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 Sterling Integrator 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 `SapSuite_<node>.ext` in the Sterling Integrator JNDI tree.

To view SAP Suite instances:

1. Login to Sterling Integrator.
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<instance>_SapSuite_node1	com.sterlingcommerce.woodstock.services.sapsuite.SapSuiteServerImplcom.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>\bin` directory.
3. Type the following command and press Enter.
 - For UNIX: `<siroot>/bin/startSAPController.sh`
 - For Windows: `<siroot>\bin\startSAPControllerWindowsService.cmd`

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: `<siroot>/bin/stopSAPController.sh`

The file, sapcontroller.pid is deleted from the <siroot> directory indicating that SAP Controller Server is stopped.

- Windows: <siroot>\bin\stopSAPControllerWindowsService.cmd

You can also stop the SAP Controller service from the Windows Services dialog. Sterling Integrator

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 Sterling Integrator 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 <siroot>/logs directory. After you register the SAP Controller in Sterling Integrator, 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 <siroot>\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 Sterling Integrator JNDI context (default). The exit code of the script is 0 if the SAP Controller server is running and registered in the Sterling Integrator 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
-s	(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:

```
(15 * * * * <si_root>/install/bin/checkSAPController.sh)
#!/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 - InstallSAPControllerWindowsService.cmd

1. Open command prompt.
2. Type uninstallSAPControllerWindowsService.cmd and press **Enter**.
3. Open InstallSAPControllerWindowsService.cmd.in in an Editor. For example, notepad.
4. Search for the lines -

```

REM Java heap memory settings

echo -Xms256m >> %PARAM_FILE%

echo -Xmx1024m >> %PARAM_FILE%

```

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 `installSAPControllerWindowsService.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 - `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. Add Java 64-Bit-Option “-d64” as follows -

```
SAPCONTROLLER_JVMARGS="-d64 -Xms256m -Xmx1024m"
```

4. Save the file and run `setupfiles.sh`.

Windows script - `InstallSAPControllerWindowsService.cmd`

1. Open command prompt.
2. Type `uninstallSAPControllerWindowsService.cmd` and press **Enter**.
3. Open `InstallSAPControllerWindowsService.cmd.in` in an Editor. For example, `notepad`.
4. Search for the lines -

```
REM Java heap memory settings
echo -Xms256m >> %PARAM_FILE%
echo -Xmx1024m >> %PARAM_FILE%
```

5. Add the following line before these lines -

```
echo -d64 >> %PARAM_FILE%
```

6. Save the file and run `setupfiles.cmd`.
7. Run `installSAPControllerWindowsService.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:

```
71,300 lines x 1,063 byte/line x 2 bytes/char = 151,583,800 bytes
```

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 Sterling Integrator pass phrase is recognized when the SAP Controller starts.

To modify the security.properties file:

1. Open the properties/security.properties file.
2. At the end of the file, add the following line -

```
enc_pass=<encrypted_gis_passphrase>
```

where <encrypted_gis_passphrase> is the encrypted pass phrase of Sterling Integrator.

3. Save and close the file.

Note: Use the encrypt_string.sh tool to get the encrypted version of your Sterling Integrator 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* and *Configuring an Outbound Route*.
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* and *Configuring an Outbound Cross-reference*.

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 Sterling Integrator

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.

ExampleSAPALEIDocInbound.bp

Service Editor - Send IDoc to SAP via ALE

Name: Send IDoc to SAP via ALE

Config: **ExampleGPMSSAPSuiteALEInboundIDoc**

SAP Suite adapter instance used to send the IDoc to SAP

Message To Service | Message From Service

Output Msg: Obtain Process Data first, then Messages

Message Name: SapSuiteInputMessage

Name	Value	Use XPATH
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		<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"/>
Synsr	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 Sterling Integrator 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

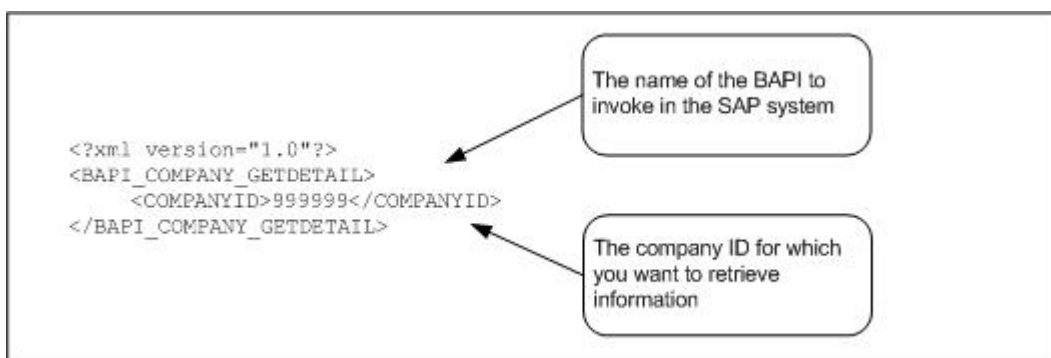
Specifies the integration mode for interacting with the SAP system.

(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 SAPSuiteInboundBAPI

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"/>
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"/>

(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"/>
Gwsevr		<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"/>
RFCFilter	RFC	<input type="checkbox"/>
RFCList		<input type="checkbox"/>

Indicates synchronous BAPI mode.

(Screen 3 of 5)

RFCList			<input type="checkbox"/>
RFCModuleName	BAPI_COMPANY_GETDETAIL	The name of the BAPI to invoke in the SAP system	<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"/>
SoLimitDelayTime	120		<input type="checkbox"/>
SyncWait	No		<input type="checkbox"/>
SyncWaitTimeout	0		<input type="checkbox"/>

(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 SCIObjectID="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.

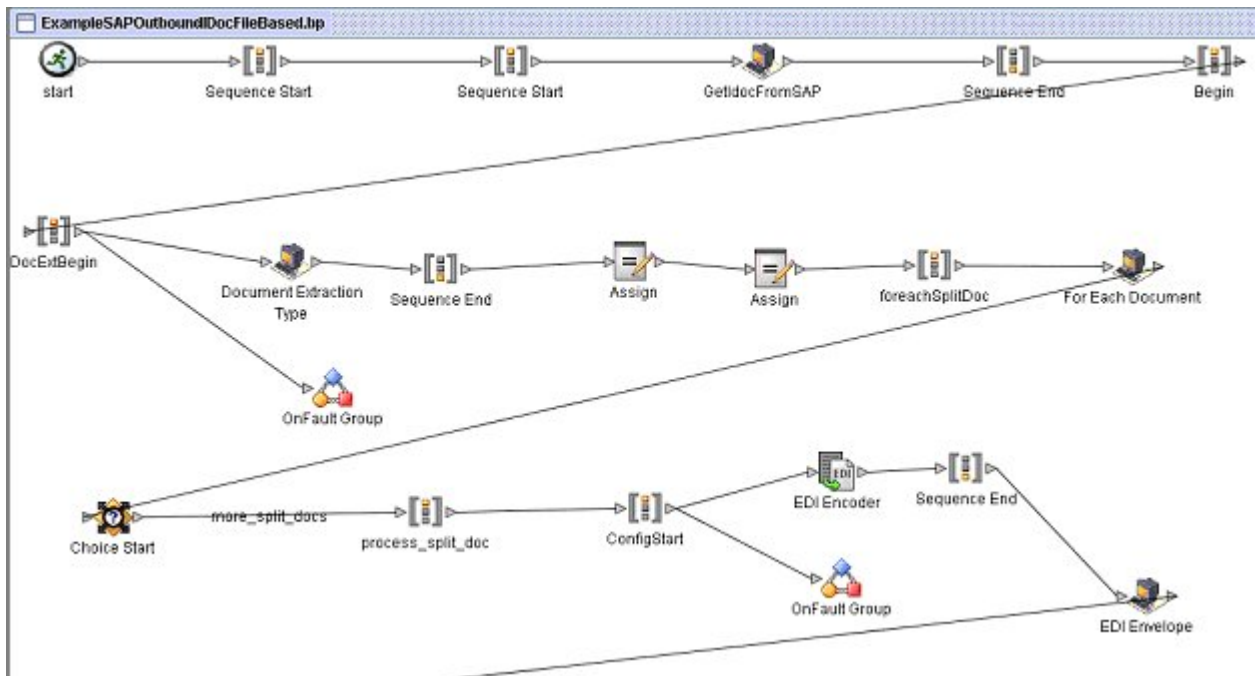
The following example illustrates an SAP Suite adapter configuration used for receiving a file-based IDoc from an SAP system.

▶ ExampleGPMSAPOutboundFileBasedIDoc

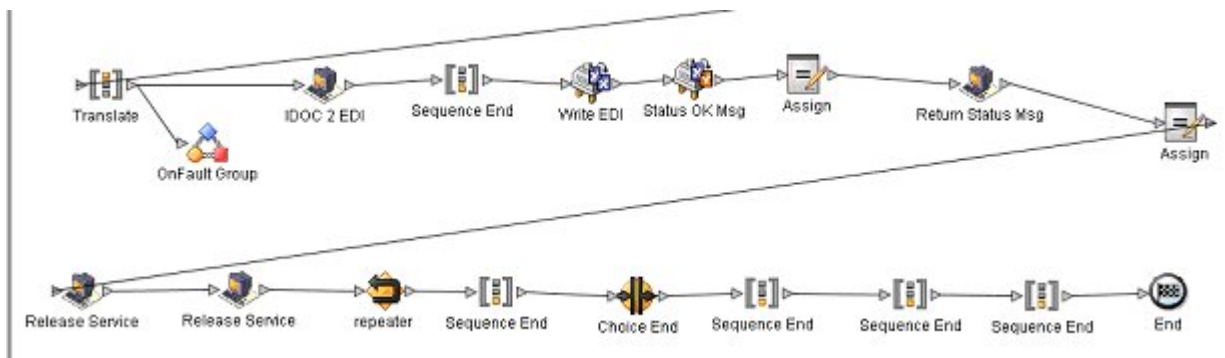
Service Settings

Service Type	SAP Suite Adapter	
Description	Configuration to include in documentation	
System Name	ExampleGPMSAPOutboundFileBasedIDoc	
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.

```

<process name="SAPOutboundIDoc">
  <rule name="more_split_docs">
    <condition>DocExtract.DONE = 0</condition>
  </rule>
  <sequence name="Sequence Start">
    <sequence name="Sequence Start">
      <operation name="GetIdocFromSAP">
        <participant name="FTPSend"/>
        <output message="FTPOut">
          <assign to="xport-ftp-document" from="RFCTriggerParams/
OrigPathAndName/text()"/>
          <assign to="xport-ftp-host" from="RFCTriggerParams/Ashost/text()"/>
          <assign to="xport-ftp-mode" from="'GET'"/>
          <assign to="xport-ftp-port" from="21"/>
          <assign to="xport-tp-authfile">Enter_FTP_Authfile_Path</assign>
        </output>
        <input message="FTPIn">
          <assign to="." from="*" />
        </input>
      </operation>
    </sequence>
  </sequence>
  <sequence name="Begin">
    <sequence name="DocExtBegin">
      <operation name="Document Extraction Type">
        <participant name="DocumentExtractionService"/>
        <output message="DocumentExtractionTypeInputMessage">
          <assign to="BatchLikeDocuments">YES</assign>
          <assign to="DocExtractMapList">sortIDoc</assign>
          <assign to="." from="*" />
        </output>
        <input message="inmsg">
          <assign to="." from="*" />
        </input>
      </operation>
      <onFault code="Unrecognized Data">
        <sequence name="Sequence Start">
          <assign name="Assign" to="PrimaryDocument" from="unrecognized/@*" />
        </sequence>
      </onFault>
    </sequence>
  </sequence>

```

Gets the IDoc from the SAP system.

Batches the incoming IDocs for each trading partner.


```

<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>

```

Loops through each batch of IDocs and processes each document.

Determines which envelope services need to run based on the AcceptorLookup Alias, SenderID, and ReceiverID.

```

        <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>

```

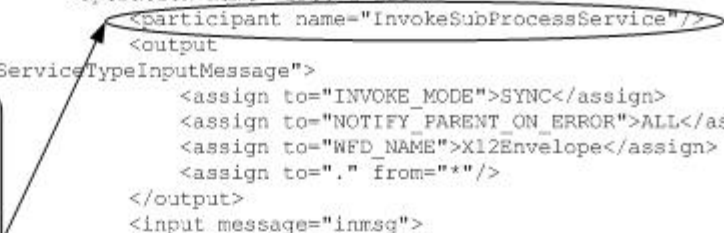
Determines which EDI enveloping business process needs to run to apply the EDI envelopes.

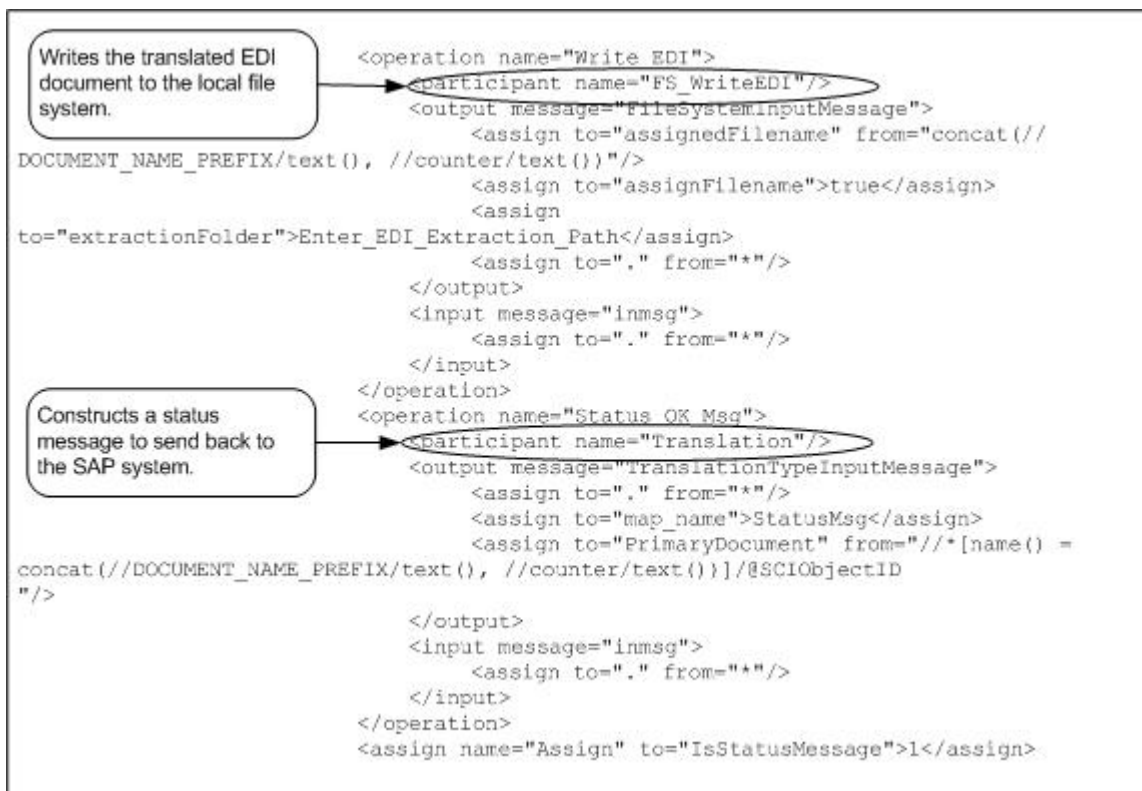
```

<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>
            </onFault>
          </sequence>
        </onFault>
      </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.





```

<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_DS40 9000000000000692334      24      3      Control information of EDI subsystem OK
EDI_DS40 9000000000000692334      06      3      Translation OK
EDI_DS40 9000000000000692334      08      3      Syntax check OK
EDI_DS40 9000000000000692334      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.

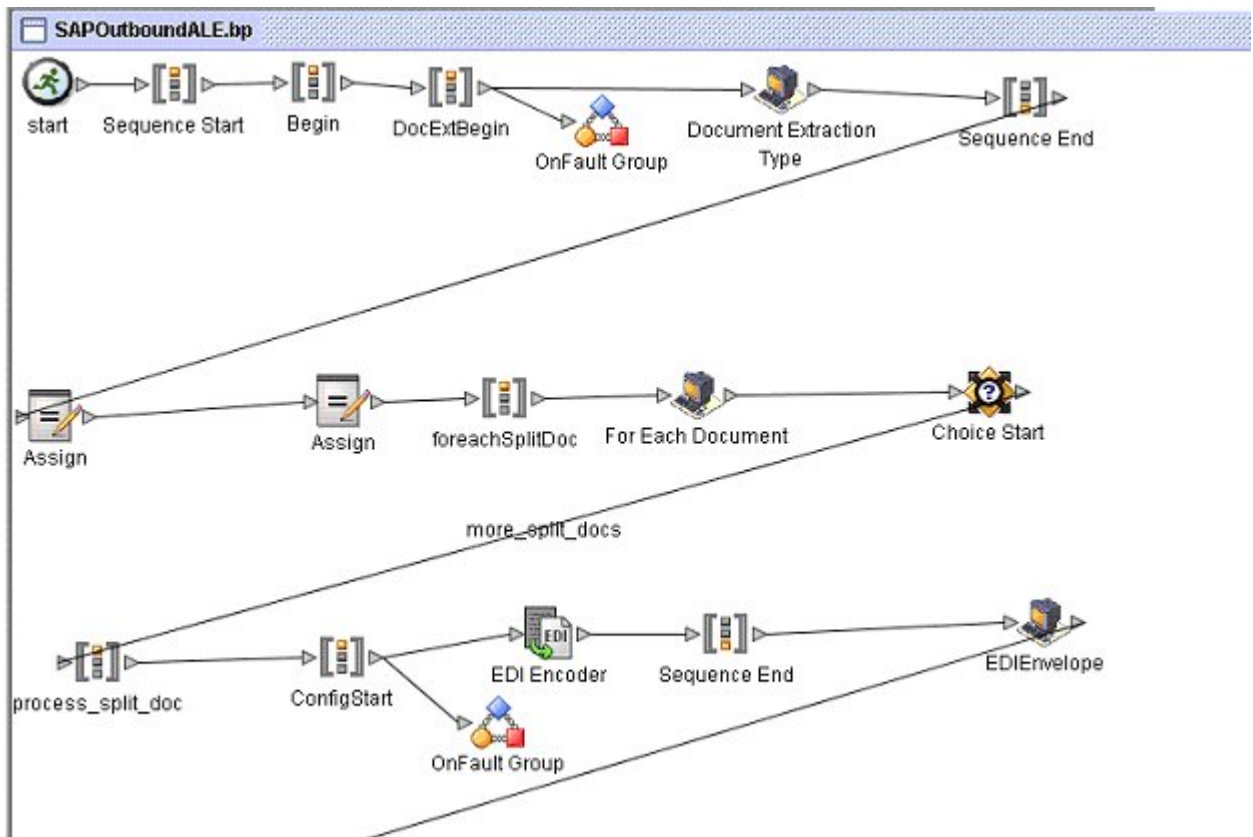
The following example illustrates an SAP Suite adapter configuration used for receiving an IDoc from an SAP system using ALE Technology.

▶ **ExampleGPM SAPOutboundALE**

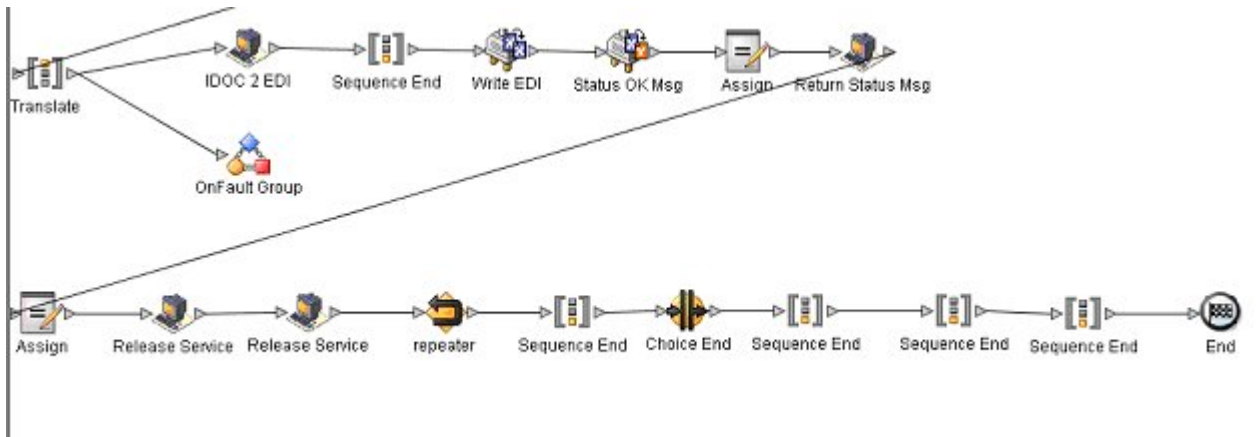
Service Settings

Service Type	SAP Suite Adapter	
Description	Configuration to include in documentation	
System Name	ExampleGPM SAPOutboundALE	
Group	None	
SAP Integration Mode	ALE based IDoc	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	
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	Business process invoked when an ALE IDoc is received from an SAP system
Program ID	SERVER_ALE_TEST	
Outbound Process to start	SAPOutboundALE	

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.

```

<process name="SAPOutboundALE">
  <rule name="more_split_docs">
    <condition>DocExtract.DONE = 0</condition>
  </rule>
  <sequence name="Sequence Start">
    <sequence name="Begin">
      <sequence name="DocExtBegin">
        <operation name="Document Extraction Type">
          <participant name="DocumentExtractionService"/>
          <output message="DocumentExtractionTypeInputMessage">
            <assign to="BatchLikeDocuments">YES</assign>
            <assign to="DocExtractMapList">sortIDoc</assign>
            <assign to="." from="**"/>
          </output>
          <input message="inmsg">
            <assign to="." from="**"/>
          </input>
        </operation>
        <onFault code="Unrecognized Data">
          <sequence name="Sequence Start">
            <assign name="Assign" to="PrimaryDocument" from="unrecognized/@**"/>
          </sequence>
        </onFault>
      </sequence>
      <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>
      </sequence>
    </sequence>
  </sequence>

```

Batches the incoming IDocs for each trading partner.

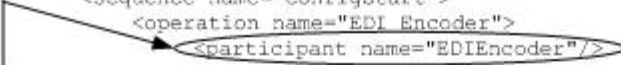
Loops through each batch of IDocs and processes each document.

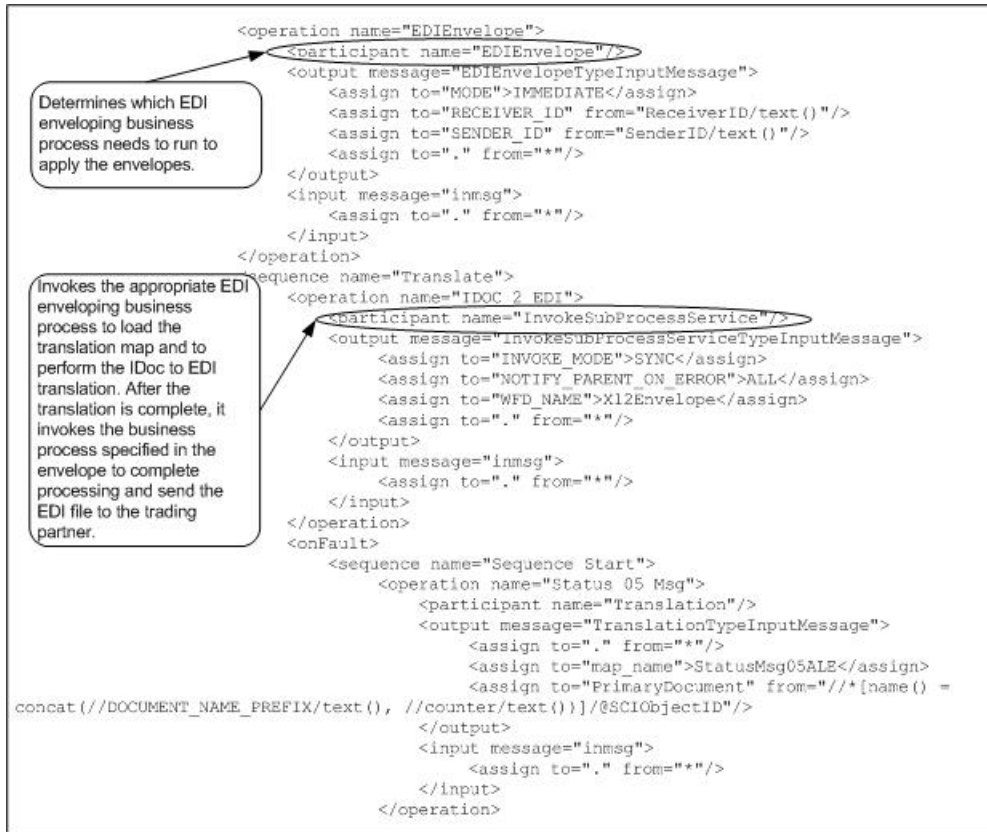

```

    <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>
          <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())]/@SCIOobjectID"/>
                </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.



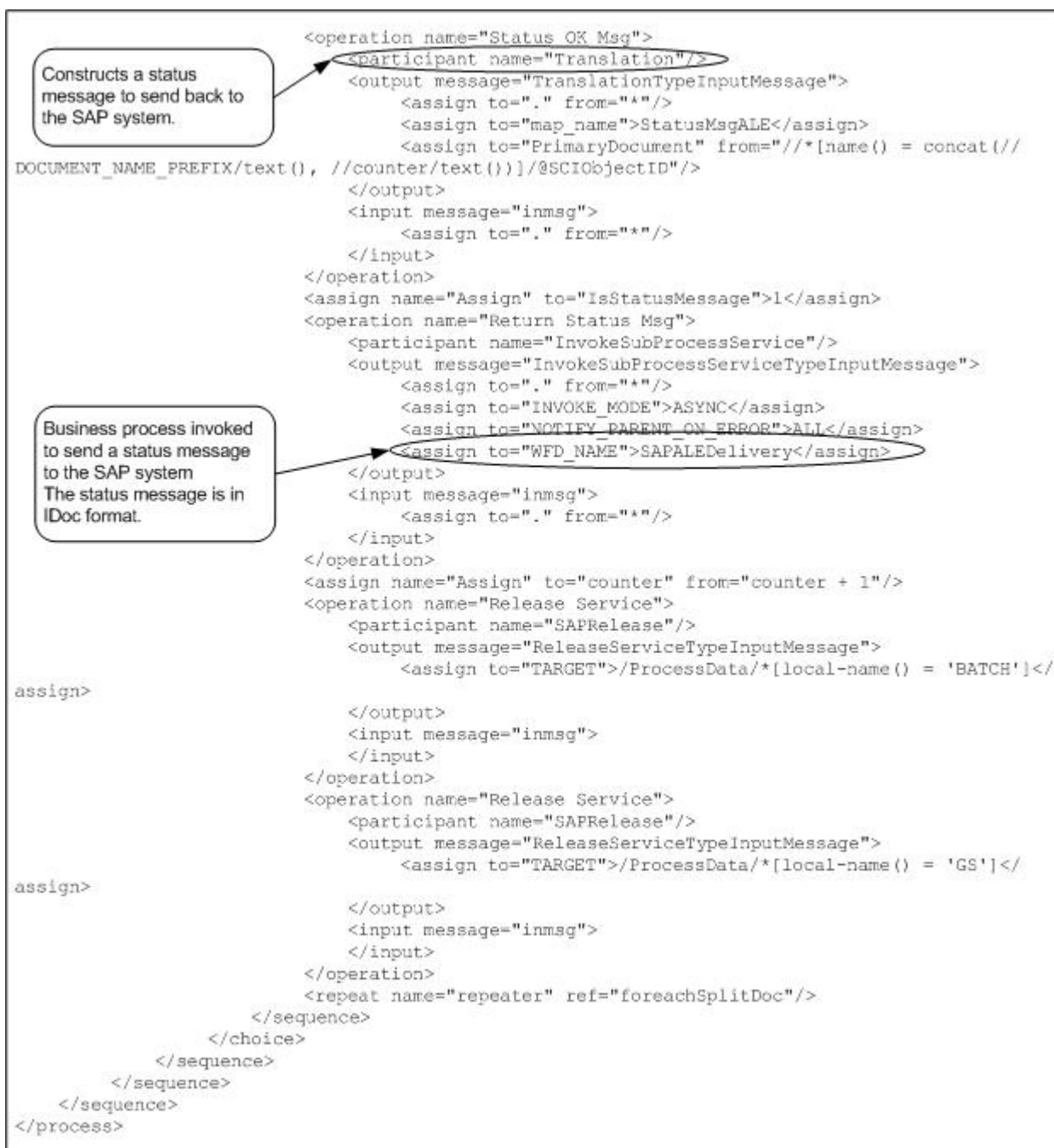


```

<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/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      Translation OK

S
E2STATS001
  900      000003000000 EDI_DS40 900000000000006922012004040804185708
          Sterling SAPSuite      Syntax check OK

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.

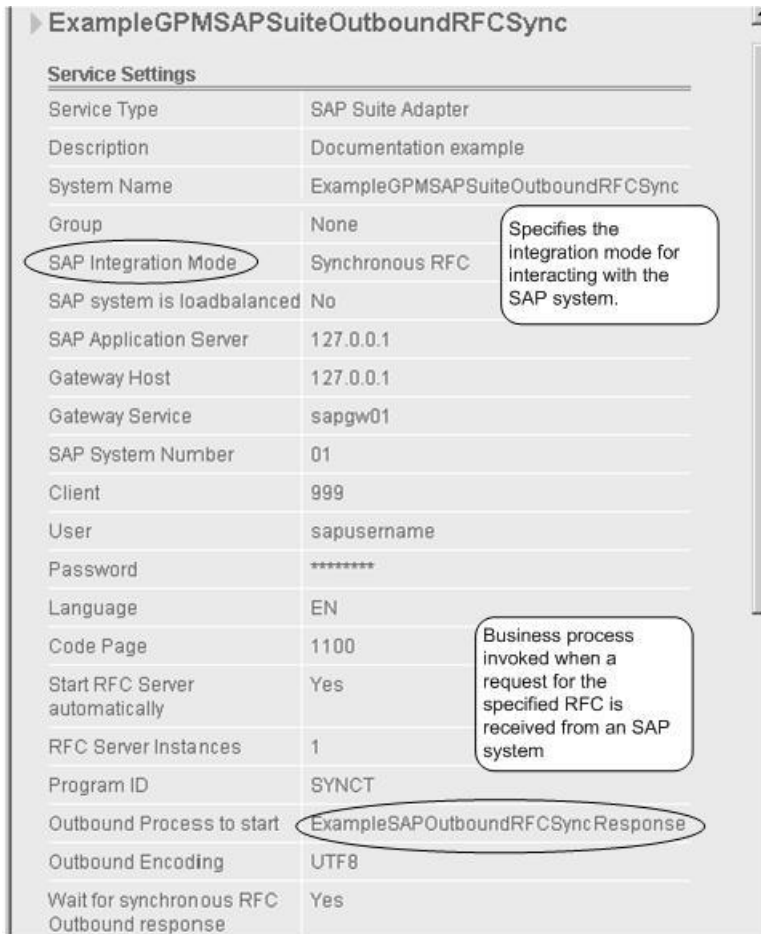
```

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.



(Screen 1 of 2)

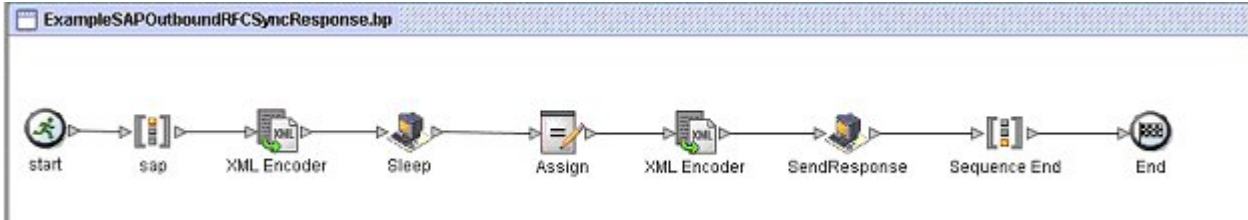
Response timeout (seconds, 0=unlimited)	120	
Delete TIDs automatically	No	
Delete TIDs after (days)	30	
Register Remote Function Calls (BAPI or RFC Mode only)	Enter RFCs offline	Option for registering RFCs
RFC List (comma separated)	Z_TRIGGERSI	The name of the RFC module to invoke
Filter online RFC list 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.

```

<process name="ExampleSAPOutboundRFCSyncResponse">
  <sequence name="sap">
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="." from="**"/>
        <assign to="." from="Result/PrimaryDocument"/>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="output_to_process_data">YES</assign>
        <assign to="root_element">ResponseDoc</assign>
      </output>
      <input message="inmsg">
        <assign to="RequestDoc" from="**"/>
      </input>
    </operation>
    <operation name="sleep">
      <participant name="SleepService"/>
      <output message="Xout">
        <assign to="MSEC_SLEEP_INTERVAL" from="1000"/>
      </output>
      <input message="Xin">
        <assign to="." from="**"/>
      </input>
    </operation>
    <assign to="Z_TRIGGERSI/RES" from="'My Response'"/>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="node">process_data_to_document</assign>
        <assign to="root_element">Z_TRIGGERSI</assign>
        <assign to="XPath" from="'Z_TRIGGERSI/node()'"/>
        <assign to="Z_TRIGGERSI" from="Z_TRIGGERSI/node()"/>
      </output>
      <input message="inmsg">
        <assign to="." from="**"/>
      </input>
    </operation>
    <operation name="SendResponse">
      <participant name="ExampleGPMSSAPSuiteOutboundRFCSync"/>
      <output message="SAPOut">
        <assign to="ServerResponse" from="'1'"/>
        <assign to="." from="**"/>
      </output>
      <input message="SAPIn">
        <assign to="." from="**"/>
      </input>
    </operation>
  </sequence>
</process>

```

Delay to simulate some activity to create a response document

SAP suite adapter instance used to send a response to the RFC back to the SAP system

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* 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.

Script Adapter

The following table provides an overview of the Script adapter:

System name	BSF
Graphical Process Modeler (GPM) category	All Services
Description	Enables Sterling Integrator 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://http://jakarta.apache.org/bsf/ .
Platform availability	All supported Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator by allowing scripts to access java objects that are normally only available to Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator. 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 Sterling Integrator is installed. Record the path and name of the directory.
2. Shut down Sterling Integrator 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 Sterling Integrator. You can now use this adapter in Sterling Integrator.

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*.
4. Test the Script adapter configuration. For information, see *Testing the Script Adapter Configuration*.
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.

Field	Description
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");
id1 = siWfd.getIDForName("zze");
siIwf.setWorkflowDefId(id1);
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);
```

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 <code>get</code> or <code>put</code> 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 Sterling Integrator 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. 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 Sterling Integrator 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>. Debug information for this adapter is located at: 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*.

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.

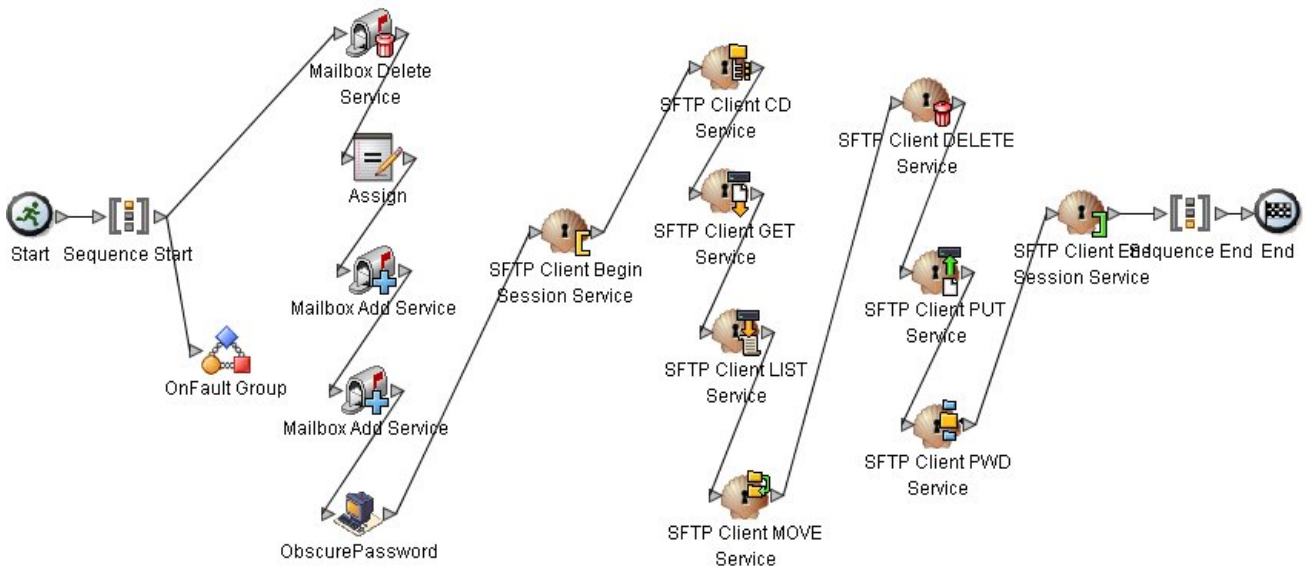
3. Click **Edit**.
4. Specify field settings:

Field	Description
Name	Name this adapter will have in Sterling Integrator
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. <hr/> <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
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.
Local 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 an SSH channel to the remote SFTP server. Optional. Valid values are:(empty) - system selects an available port0 - system selects an available port $nnnn$ - use specified port $nnnn$, for example 9012 $nn-yy$ - use a port in the range of nn to yy , for example 462-863 would use a port in the range of 462 to 863, inclusiveMultiples of the above values can be specified, separated by commas, for example 9012, 462-863, 4925.
Use proxy server	<p>Valid values are:</p> <ul style="list-style-type: none"> • Yes • No
Proxy hostname	A valid proxy server hostname or IP address to use as a gateway.
Proxy port	Listen port number for the proxy server.

Field	Description
Proxy Retries	The number of attempts to connect to the proxy server if the connection is unsuccessful.
Use proxy basic authentication	Valid values are: <ul style="list-style-type: none"> • Yes • No
Proxy username	Username to connect to the proxy server.
Proxy password	Password to connect to the proxy server.

Business Process Example of All SFTP Client Services

To help you get started using the SFTP Client adapter and SFTP Server adapter, Sterling Integrator 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>
<!-- 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>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>

```

```

    <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>
        <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. Sterling Integrator 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 Sterling Integrator 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</p>

	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.
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	Returned status values: <ul style="list-style-type: none"> • 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 - Sterling Integrator uses those values.
- No - Is ProfileId specified?
 - Yes - Sterling Integrator 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 Sterling Integrator
Description	Description of service

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. <hr/> <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
PreferredAuthenticationMethod	<p>Method used to authenticate users. Required. Valid values are:password, public key</p> <hr/> <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> <hr/> <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> <hr/> <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 data-bbox="852 1738 1469 1890"> <assign to="ResponseTimeout"> 300</assign> <assign to="RetryDelay">30</assign> <assign to="ConnectionRetries"> 5</assign> </pre>

Field	Description
KnownHostKeyId	<p>Public key used to authenticate remote SFTP servers to the Sterling Integrator SFTP Client. Internal value is GUID. Required.</p> <hr/> <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: (empty) - system selects an available port0 - system selects an available port <i>nnnn</i> - use specified port <i>nnnn</i>, for example 9012nn-yy - use a port in the range of nn to yy, for example 462-863 would use a port in the range of 462 to 863, inclusiveMultiples of the above values can be specified, separated by commas, for example 9012, 462-863, 4925.</p> <hr/> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
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 <hr/> <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> <hr/> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
ProfileId	<p>SSH Remote profile identification. Optional. Valid value is any valid profile ID.</p>

Field	Description
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>
ResponseTimeout	<p>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.</p> <p>Note: The value entered for this parameter overrides the setting in the SSH Remote Profile configuration.</p>
RetryDelay	<p>Number of seconds the adapter will wait before retrying. Optional. Valid value is any numeric value.</p> <p>Note: The value entered for this parameter overrides the Retry Delay setting in the SSH Remote Profile.</p>
SFTPClientAdapter	<p>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.</p>

Field	Description
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

If you provided a ProfileId in the Input from Business Process to Service, the SFTP Client Begin Session service uses the profile information you configured in the SSH Remote Profile to output the profile information to the process data. The following example illustrates the elements of an SSH remote profile in an XML ProcessData:

```
<ProcessData>
...
<TPProfile>
  <ProfileName>...</ProfileName>
  <RemoteHost>...</RemoteHost>
  <RemotePort>...</RemotePort>
  <KnownHostKey>...</KnownHostKey>
  <RemoteUser>...</RemoteUser>
  <PreferredAuthenticationType>...</PreferredAuthenticationType>
  <UserIdentityKey>...</UserIdentityKey>
  <Compression>...</Compression >
  <ConnectionRetryCount >...</ConnectionRetryCount>
  <RetryDelay>...</RetryDelay>
  <ResponseTimeout>...</ResponseTimeout>
  <PreferredCipher>...</PreferredCipher>
  <PreferredMAC>...</PreferredMAC>
  <LocalPortRange>...</LocalPortRange>
  <Directory>...</Directory>
</TPProfile>
...
</ProcessData>
```

A Business Process can also use the ProcessData element <Directory> in the SFTP Client CD Service to change the configured directory.

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.

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.

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 Sterling Integrator 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 Sterling Integrator 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	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	Returned status values: <ul style="list-style-type: none"> • 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

Parameter	Description
	<ul style="list-style-type: none"> • 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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • 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

Parameter	Description
	<ul style="list-style-type: none"> • 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, Sterling Integrator 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 Sterling Integrator 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	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 Sterling Integrator is restarted.
Business process context considerations	None
Returned status values	Returned status values: <ul style="list-style-type: none"> • 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 Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator 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	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	Returned status values: <ul style="list-style-type: none"> • 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.
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

Parameter	Description
	<ul style="list-style-type: none"> • 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
Protocol	SFTP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost
Username	username
RemoteFile	filename

SFTP Client LIST Service

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 Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator 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	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	Returned status values: <ul style="list-style-type: none"> • 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • 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

Parameter	Description
	<ul style="list-style-type: none"> • 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 Sterling Integrator 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, Sterling Integrator 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 Sterling Integrator 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</p>

	service. It may be used to put a document that is returned from an SFTP Client GET 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	Returned status values: <ul style="list-style-type: none"> • 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	<p>The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are:</p> <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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • 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>
[[Insert End Session]]
```

SFTP Server Adapter

The SFTP Server adapter enables external SFTP clients or SCP clients to put files into, or get files from, a mailbox in this application or to a physical file system on the server. The SFTP Server adapter:

- Uses Perimeter services.
- Uses the Mailbox subsystem or the physical file system directory as its repository (virtual roots).
- Uses routing rules for items placed in Mailbox to trigger a business process, or if items are placed in a directory on the file system, you can identify a business process to be invoked each time a new message or file is received.
- 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 put files into, or get files from, a mailbox in this application or to a physical file system on the server.
Usage example	A trading partner uses an SFTP client to retrieve a business document from a mailbox. The SFTP Server adapter receives and processes the trading partner request.
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 your application. To enable the SFTP Server adapter, you must first disable DemoAllSFTPServerAdapter or change the port assignment.
Requires third party files?	No

Platform availability	All supported platforms for this application
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 the application by a firewall. Refer to the perimeter services documentation for details on installing and running that component.
Initiates business processes?	<p>The SFTP Server adapter can:</p> <ul style="list-style-type: none"> • Can initiate business processes if the Payload Repository is a File System. You can configure the adapter to invoke a specific business process each time a message or file is placed in the home directory. • Does not initiate business processes if the Payload Repository is a mailbox. However, mailbox activities can trigger routing rules.
Invocation	This adapter is not invoked from a business process.
Business process context considerations	None
Returned status values	Not applicable
Restrictions	<p>Restricted to platforms that support Java SDK version 1.5 and above. Transfer resumption (for mailboxes) is disabled by default. To enable transfer resumption and listing documents that are in the staging area, edit the sftp.properties file (located at <install_dir>/properties/sftp.properties.in) to set listStagedDocuments = True.</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 time frame). 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:</p> <ul style="list-style-type: none"> • Complete documents in the target mailbox. • Partial documents in the staging area. Partial documents are assigned to a particular user. The system only displays partial documents to the user to whom they are assigned. <p>If two documents with the same name exist in both the mailbox and the document staging area, only the partial</p>

	<p>document in the staging area is displayed in response to a list request.</p> <p>The home directory for SFTP is a virtual root mailbox in the application or a path and directory specified on a physical file system on the server. The mailbox can include both extractable and nonextractable messages. When the SFTP Server adapter accesses the home directory, only extractable messages are displayed.</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>
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 MailboxLoginWithoutVirtualRootPermission. With this</p>

	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	Default
Testing considerations	<p>At application startup, attempt to access the SFTP server using a supported SFTP client with the configured IP address and port. Debug information can be found in the SFTP logs. Select Logging Level from the following:</p> <ul style="list-style-type: none"> • Error – Errors only • Communication Trace – Errors, requests from clients, and responses from the Server adapter, including ACL violations • All – 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 the 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 the application
Description	Description of adapter
Environment	<p>The Environment field is displayed only in a cluster setup. Required.</p> <p>Select the node in which the adapter should be deployed. If you do not select any node, all nodes will be selected by default and the adapter will start on the node that will be started first. The SFTP session must be node-specific. When the traffic starts, if the required node is not the default node, the adapter will be disabled. Restart traffic by editing the adapter configuration and selecting the correct node or container node.</p>
Select a Group	None – Do not include this configuration in a group.

Field	Description
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 (default). • SFTP. • SCP. 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. <hr/> 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. <hr/> 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.
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.

Field	Description
Session Timeout (seconds)	<p>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.</p> <hr/> <p>Note: If the timeout is reached during a transfer, the session will be closed immediately after the transfer completes.</p>
Idle Connection Timeout (minutes)	<p>The number of minutes after which the server adapter closes the TCP connection if the client/connection is idle for that length of time. Optional.</p>
Resumption Timeout (hours)	<p>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.</p>
Compression	<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. Optional. Valid values:</p> <ul style="list-style-type: none"> • None • ZLIB
PreferredCipher	<p>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:</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
PreferredMAC	<p>The MAC the server prefers to use for stream encryption. Optional. Valid values are:</p> <ul style="list-style-type: none"> • hmac-sha1 (default) • hmac-md5
Required Authentication	<p>Specifies the type of authentication required for the adapter. Required. Valid values are:</p> <ul style="list-style-type: none"> • Password or Public Key (default) • Password • Public Key • Password and Public Key

Field	Description
	<p>Note: If an application user account is associated with multiple public keys (SSH Authorized User keys), any of the corresponding private keys can be used to log into the SFTP Server adapter.</p>
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.
Payload Repository	<p>Whether files or messages will be stored in a mailbox or a physical file system on the server. Required. Valid values are:</p> <ul style="list-style-type: none"> • Mailbox (default) - If you want to restrict user access to specific mailboxes, see <i>Mailbox Features, Creating Virtual Roots</i> documentation. • File System - If you want to restrict user access to specific file system folders and subfolders, see the <i>Configuring an File System Virtual Root</i>.
Document Storage Type	<p>Select whether documents will be stored on the file system, the database, or the system default. Required. Valid values are:</p> <ul style="list-style-type: none"> • File System (default) – Default value when the application is installed, but it can be changed. Contact your system administrator to see if the default has been changed. • Database – Body of the request document will be stored in the database. • System Default – If your system administrator has changed the installed default of File System, this ensures that the correct location is used.
Add Policy Type	If you want to apply an existing policy to this instance, select the plus sign.
Select policy type	<p>Select one of the adapter policy types:</p> <ul style="list-style-type: none"> • Lockout Policy • Bandwidth Limiting Policy • Command Limiting Policy • Data Limit Policy

Field	Description
Select Policy	Select from the list. Policy must have already been created.
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.
Should users start in the directory that matches their user name upon login?	Places the user, upon logging in, into a directory (mailbox in the application) that corresponds to his or her user ID. Valid values are: <ul style="list-style-type: none"> • Yes – Upon login, the user is automatically placed in a directory that matches his or her user ID. If such a directory is not available, the user is placed in the virtual root directory. This option allows Connect:Enterprise UNIX customers to run production scripts that require each user to be placed into directories that correspond to user ID. Caution: Do not select Yes if there is any chance that users of your application might have user IDs that differ only by case (example: jsmith and JSmith). Unlike user IDs, mailbox names in this application are not case-sensitive. • No – The user is placed in the virtual root directory.
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 dddhhmm.
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**.

Note: The SFTP Server adapter will stop all active connections and shut down automatically when the database connection goes down. The SFTP Server adapter restarts after the database connection is up and running.

Correlations and Document Tracking

The following table details the correlations available from the SFTP Server adapter for document tracking:

Key	Values
ACTION	Get, Put

Key	Values
Direction	Inbound, Outbound
Protocol	SFTP or SCP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost
Username	username

Adding Policies to the SFTP Adapter

You can apply adapter policies to the SFTP adapter. You can define Lockout, Bandwidth Limiting, Command Limiting, and Data Limit policies from the Admin Console UI (Deployment > Adapter Utilities > Policies). For more information on creating Adapter Polices, see *Adapter Polices*.

Using Multiple SSH Keys for a single user

You can associate multiple authorized user (SSH) keys with a single Application user. You can share a single user ID with multiple trading partners who use different (private) SSH keys to authenticate to the SFTP Server adapter.

Correlations and Document Tracking

The following table details the correlations available from the SFTP Server adapter for document tracking:

Key	Values
ACTION	Get, Put
Direction	Inbound, Outbound
Protocol	SFTP or SCP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost
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 Options That Are Supported

Incoming Packet Types:

- INIT: Initialize the protocol (in).
- VERSION: Specify the version of the protocol that agrees with both client and server (out).
- OPEN: Opens a file for reading and/or writing.
- CLOSE: Closes a file.

- READ: Reads data from a file.
- WRITE: Writes data to a file. Also supports the following as part of WRITE: Append (APPEND), Exclusive (EXCL), Create (CREAT), Truncate (TRUNC).
- OPENDIR: Opens a directory for reading.
- READDIR: Reads files and directories from within a directory.
- REMOVE: Removes a file.
- RENAME: Renames a file (will not support directories).
- STAT: Status of a file (does not follow links).
- LSTAT: Status of a file (follows links).
- FSTAT: Status of an open file.
- REALPATH: Returns the absolute canonical path of a file or directory.

Outgoing Packet Types:

- STATUS: The result of a command.
- HANDLE: A handle to a file.
- DATA: Data from a READ.
- NAME: The name of a file or directory from a READDIR.

SFTP Server Options That Are Not Supported

Incoming Packet Types:

- SETSTAT: Changes the status of a file.
- FSETSTAT: Changes the status of an open file.
- READLINK: Reads a symbolic link.
- SYMLINK: Creates a symbolic link.
- EXTENDED: Sends an extended command.

Outgoing Packet Types:

- EXTENDED:_REPLY: Replies to an extended command.

SFTP Server Adapter Starts/Restarts

When restarting the SFTP Server Adapter, allow for the other necessary processes to restart. For example, after a database shutdown, there are associated processes which need to go down before the SFTP Server shuts down. This is also applicable in the case of a database start up. Pre-requisite processes for the SFTP Server start before the SFTP Server starts up. It takes several minutes for all of pre-requisite services to restart. The amount of time to restart is highly variably by environment.

File System Virtual Root for SFTP

When you configure an SFTP adapter and the Payload Repository is defined as File System, and if you want to restrict user access to specific file system folders and subfolders, then you need to configure the file system virtual root. The file system virtual root is relative to the adapter Base Directory. The virtual root defines the point of access for each user who has permission to use the adapter. The virtual root is relative to the Base Directory.

Configuring a File System Virtual Root

Before you begin, you need to know:

- User ID that need permission to the adapter virtual root
- Path to the Base Directory

- Create a folder under the base directory which will be the virtual root

To create a new File System Virtual Root:

1. Navigate to the **Administration Menu > Deployment > Adapter Utilities > FS Virtual Root**.
2. Next to **Create a new Virtual Root**, click **Go!**
3. Select the **User ID** from the list and click **Next**.
4. Enter the path to the virtual root.

For example, if the base directory is /install_dir/install/ftpserver1, then the file system virtual root can be any folder/directory under the /install_dir/install/ftpserver1 directory.

5. Click **Finish**.

Editing a File System Virtual Root

To edit a File System Virtual Root:

1. Navigate to the **Administration Menu > Deployment > Adapter Utilities > FS Virtual Root**.
2. Use either Search or List to locate the User ID for which the virtual root needs to be edited.
3. Click **edit** next to the User ID. The User ID is displayed.
4. Click **Next**.
5. Update the Virtual Root and click **Next**.
6. Click **Finish**.

Deleting a File System Virtual Root

To delete a File System Virtual Root:

1. Navigate to the **Administration Menu > Deployment > Adapter Utilities > FS Virtual Root**.
2. Use either Search or List to locate the Virtual Root.
3. Click **delete** next to the User ID which virtual root needs to be deleted.
4. Click **OK**.
5. Review the virtual root information.
6. Click **Delete**.

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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator and in the Graphical Process Modeler (GPM).

Sterling Integrator Configuration

The following table describes the fields used to configure the Sterling Information Broker adapter in Sterling Integrator:

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 Sterling Integrator 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.
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 Sterling Integrator to use with this adapter. Optional.
xport-tp-cipher	Level of encryption Sterling Integrator is to apply to the data that flows through the socket connection. Valid values are STRONG (default), WEAK, and ALL. Optional.

Field	Description
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 Sterling Integrator interface configuration.

Before Using the Sterling Information Broker Adapter in a Business Process

If the FTP server that is communicating with Sterling Integrator 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:

- 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**.
- Search for the appropriate certificate name, using the Search or List function.
- 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.
- From the **Search** menu, select **Find**.
- In the **Find what** field, type the name of the certificate as it appears in the list of search results returned in Sterling Integrator.

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>
```

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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none">• 0 –Success• 1 – 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:

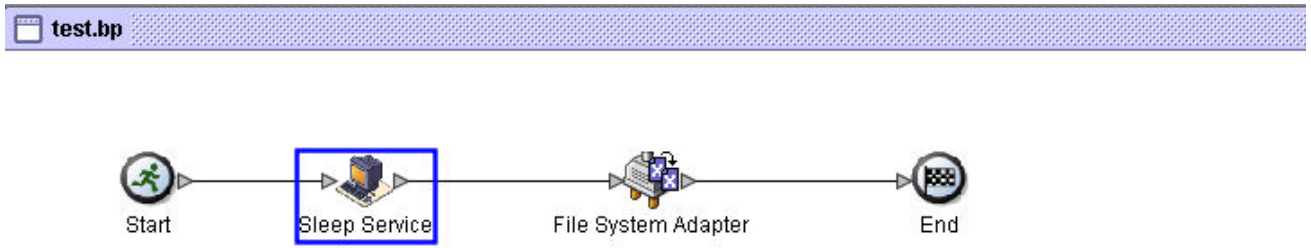
Field	Description
Config	Select the name of the service configuration from the list.
MSEC_SLEEP_INTERVAL	Duration of sleep time in milliseconds. Required.

Field	Description
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:

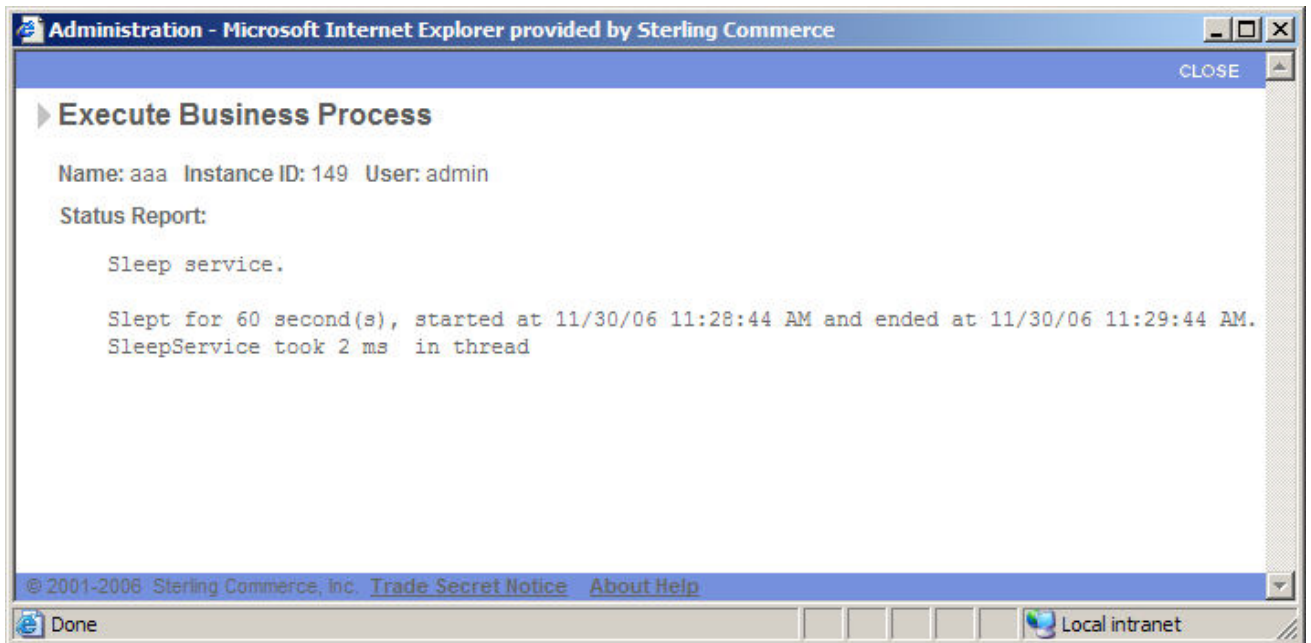
```
<process name="default">
  <sequence>
    <operation name="Sleep Service">
      <participant name="SleepService"/>
      <output message="SleepServiceInputMessage">
        <assign to="SLEEP_INTERVAL">60</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <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.

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 Sterling Integrator. It does this by enabling Sterling Integrator to mail (send) documents using SMTP, within Sterling Integrator, 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 Sterling Integrator 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 Sterling Integrator). It does this by enabling Sterling Integrator to mail (send) documents using SMTP, within Sterling Integrator, to the mail server.
Business usage	Documents and messages are sent by e-mail through the use of the SMTP Send adapter. Sterling Integrator 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 Sterling Integrator platforms
Related Services	<p>Related services include:</p> <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	<p>Values:</p> <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. 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

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.

Note: The Sterling Integrator 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.

Setting the b2b-raw-message to false defines the type of message contained in the primary document.

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"/>

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 Sterling Integrator 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*.
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 Sterling Integrator and in the GPM.

Sterling Integrator Configuration

The following table describes the fields used to configure the SMTP Send adapter in Sterling Integrator:

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 Sterling Integrator 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.
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.

Field	Description
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.
SMIME Decryption User Certificate (System Store)	Not used in this release. Leave at default.
SSL	The Secure Socket Layer (SSL) flag that turns on SSL socket negotiation. Optional. Valid values: <ul style="list-style-type: none"> • SSL_NONE – SSL socket negotiation is turned off. Default. • SSL_MUST – SSL socket negotiation is turned on. Note: Can also be configured in the GPM.
Key Certificate Passphrase	Passphrase for the keystore if you are using SSL Client Authentication. Optional.
Cipher Strength	The level of encryption applied to the message that flows through the socket connection. Optional. Valid values: <ul style="list-style-type: none"> • All – Default

Field	Description
	<ul style="list-style-type: none"> • Strong • Weak <hr/> <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. Valid values: Any valid digital certificate in Sterling Integrator.</p> <hr/> <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> <hr/> <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

Message To Service | Message From Service

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
smtpnotifyDelivery	xport-smtp-notifyDelivery

SMTP Service Configuration(BPML value)	GPM Configuration(BPML value)
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	Indicates if the primary document input to the adapter is a raw or non-raw message. Valid values: <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	Delivery notification requested. Optional. Valid values: <ul style="list-style-type: none"> • Yes – (true) Requests delivery notification. • No – (false) No delivery notification requested. Default. <hr/> <p>Note: Delivery notification only available for mail servers that support this option.</p>
xport-smtp-notify-read	Read notification requested. Optional. Valid values: <ul style="list-style-type: none"> • Yes – (true) Requests read notification. • No – (false) No read notification requested. Default. <hr/> <p>Note: Read notification only available for mail servers that support this option.</p>

xport-smtp-retries	Number of times SMTP Send adapter tries to connect to the SMTP server if the connection is not successful. Optional. Valid value is an integer.
xport-smtp-retryinterval	Time delay between retrying the connection. Optional. Valid value is an integer.
xport-smtp-mailsubjectencoding	The type of encoding for the e-mail message subject. 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
xport-tp-ssloption	SSL flag that turns on SSL socket negotiation. Valid values: <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

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"/>

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.

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

```
<process name="Test_SMTP_SSL">
  <sequence name="e-mail sequence">
    <operation name="Send with copies">
```

```

<participant name="SMTP_SEND_ADAPTER"/>
<output message="Xout">
  <assign to="." from="*"></assign>
  <assign to="b2b-raw-message">false</assign>
  <assign to="xport-smtp-auth">>true</assign>
  <assign to="xport-smtp-mailBCC">stest4@co.com</assign>
  <assign to="xport-smtp-mailCC">stest3@co.com</assign>
  <assign to="xport-smtp-mailfrom">stest1@co.com</assign>
  <assign to="xport-smtp-mailhost">00.000.00.00</assign>
  <assign to="xport-smtp-mailport">25</assign>
  <assign to="xport-smtp-mailsubject">This is the subject</assign>
  <assign to="xport-smtp-mailto">stest2@co.com</assign>
  <assign to="xport-tp-cakeycert-id">server:105e37:f3be012345:-52e9
</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>
</sequence>
</process>

```

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.

```

This is a simple example of a text message.
This message is used to test the
SMTP Send adapter configuration with the non raw message format.
This entire text appears in the body of the e-mail.

```

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:

```

From: "John, Doe" <john_doe@e-mailaddress.com>
To: "Jane, Jones" <jane_jones@e-mailaddress.com>
Subject: Testing Raw Message
Date: Mon, 16 Feb 2004 15:51:44
MIME-Version: 1.0
Content-Type: text/plain charset="iso-8859-1";
This is a simple RFC822 compliance message.
The message envelope consists of the originator

```

and recipient fields, and this is the content of the message.

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		<input type="checkbox"/>
smtpRetryInterval	This is used when the primary document is internet-compliant and no	<input type="checkbox"/>
smtpSubject	tr	<input type="checkbox"/>
smtpInstitution	insitution is needed.	<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: A callout box points to the 'b2b-raw-message' row, stating: '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 insitution is needed.'

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

```

<process name="Test SMTP Using Raw Message">
  <sequence name="Raw Message">
    <operation name="e-mail Raw Message">
      <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>
```

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.

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>. • 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>.
Preconfigured?	No
Requires third party files?	Slv1_snmp4_10.jar
Platform availability	All supported Sterling Integrator platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Restrictions	See <i>Requirements</i> .

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.

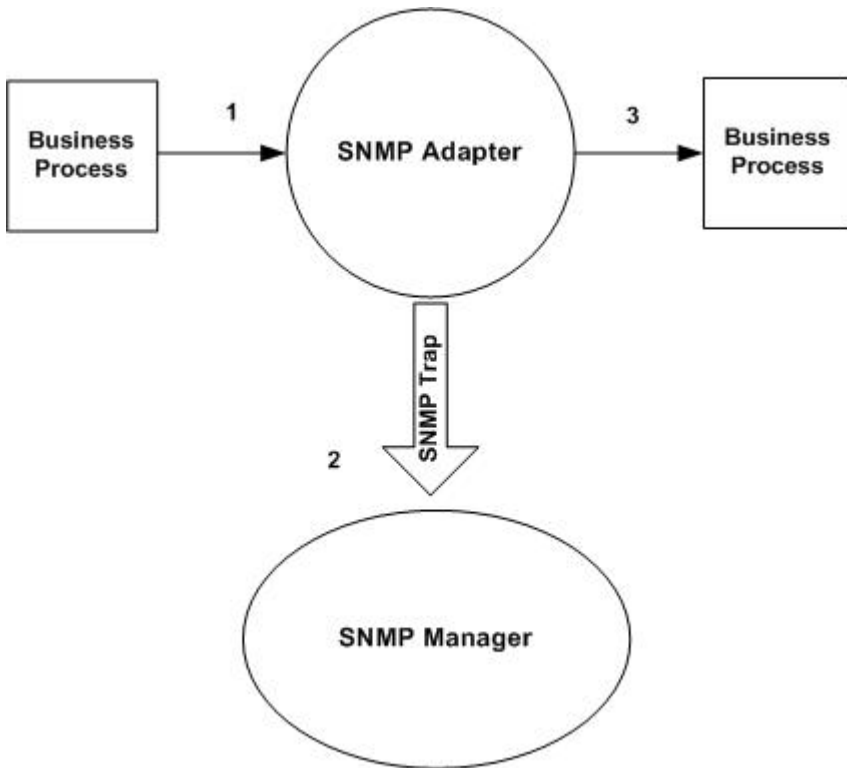
The SNMP trap generator does not require a document for input; the adapter obtains all inputs through business process parameters that are set either during adapter configuration, business process definition, or at run time.

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 Sterling Integrator 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 Sterling Integrator, 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 Sterling Integrator.

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 Sterling Integrator.
5. Sterling Integrator 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*.
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 Sterling Integrator:

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.
Trap Receivers (trapReceivers)	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. 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.
Community Name (community)	Access environment for a group of network management stations. If left blank, defaults to <code>public</code> .
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

Field	Description
	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)	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: <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 Sterling Integrator 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 Sterling Integrator. 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.

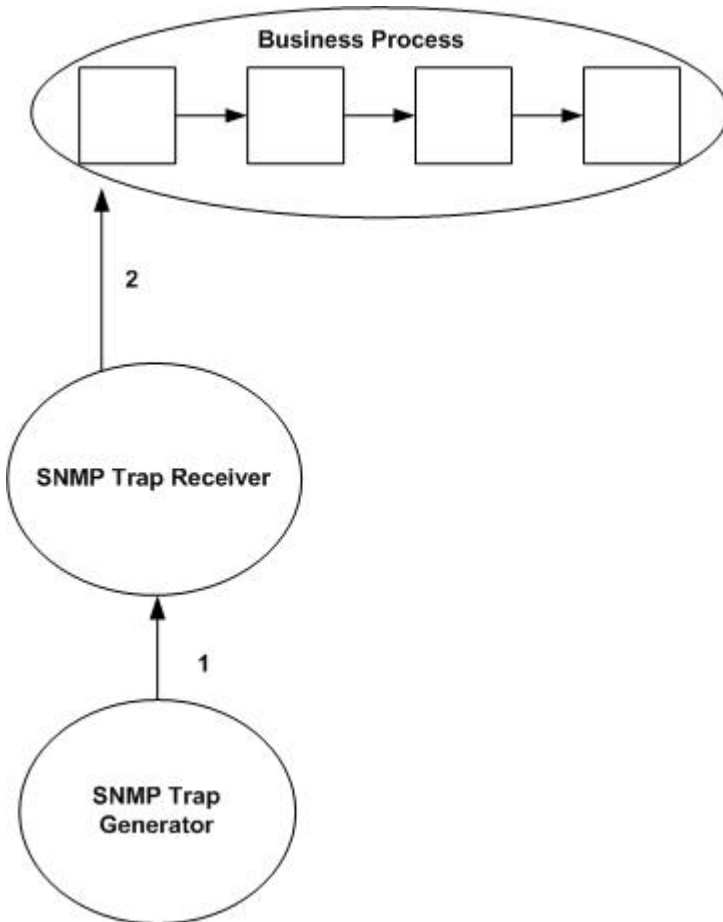
Note: Because the configuration for an SNMP trap receiver is not started within a business process, the configuration cannot be overridden.

The SNMP trap receiver operates over UDP; there is no guarantee that a trap will be received by Sterling Integrator. 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 Sterling Integrator 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 Sterling Integrator may generate a trap message to an SNMP Trap receiver in another installation of Sterling Integrator.

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.

- 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:

- Create an SNMP Trap adapter configuration. For information, see *Managing Services and Adapters*.
- Configure the SNMP Trap adapter. For information, see *Configuring the SNMP Trap Adapter As a Trap Receiver*.
- 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 Sterling Integrator:

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	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.
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	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. Optional; if left blank, no community names are excluded.

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.

SOA Inbound Message Processing Service

The SOA Inbound Message Processing service removes HTTP headers from the incoming SOAP request for both the provider and the 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 incoming SOAP request for both the provider and the 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 a consumer process to remove HTTP headers from a SOAP 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 platforms for your application
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 put after HTTP adapter service). 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.
Business process context considerations	None
Returned status values	Returned status values: <ul style="list-style-type: none"> • Success – Service is able to understand 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	Debug information can be found by observing the execution flow of the business process in the application user interface. The level of logging can be controlled through the log.properties file, just as for other log files. Also see Business Usage, Usage Example and Application Requirements in the previous sections.

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 <hr/> <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.

```
<!-- New BP for the Web Service layer -->
<process name = "WS_MessageHandler">
  <rule name="NonRMBranch">
    <condition>messageMode=1</condition>
  </rule>
  <rule name="RMBranch">
    <condition>messageMode=0</condition>
  </rule>
  <rule name="RMFaultBranch">
    <condition>reliableMode=0</condition>
  </rule>
  <rule name="NonRMFaultBranch">
    <condition>reliableMode=1</condition>
  </rule>
  <sequence name="Sequence Start">
    <assign to="incomingSOAPTransportMode">http</assign>
    <assign to="incomingSOAPTransportMode/@attr1">/assign>
    <operation name="WSConfigInfoService">
      <participant name="WSConfigInfoService_Instance"/>
      <output message="WSConfigInfoServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
<!-- New SOA Inbound Service Hndle only MIME part now -->
<operation name="SOAInboundMsgProcessingService">
  <participant name="SOAInboundMsgProcessingService_Instance"/>
  <output message="SOANewInboundTypeInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<!-- New RM Decision Service -->
<operation name="SOAInboundRMDecisionService">
  <participant name="SOAInboundRMDecisionService_Instance"/>
  <output message="RMDecisionServiceTypeInputMessage">
    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<choice name="RM Choice">
  <select>
    <case ref="RMBranch" activity="ProcessRMPart"/>
    <case ref="NonRMBranch" activity="ProcessNonRMPart"/>
  </select>
  <sequence name="ProcessRMPart">
    <operation name="Reliable Service Handler">
      <participant name="RMServiceHandler_Instance"/>
      <output message="RMSHServiceTypeInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="Reliable Service Manager">
      <participant name="RMServiceManager_Instance"/>
      <output message="RMSMServiceTypeInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  <choice name="Choice Start">

<select>
  <case ref="RMFaultBranch" activity="ProcessRMFaultPart"/>
  <case ref="NonRMFaultBranch" activity="ProcessNonRMFaultPart"/>

```



```

        </select>
        <operation name="ProcessRMFaultPart">
<participant name="HttpRespond"/>
<output message="HttpRespondServiceInputMessage">
<assign to="." from="*"></assign>
<assign to="doc-has-headers">false</assign>
</output>
<input message="inmsg">
<assign to="." from="*"></assign>
    </input>
</operation>
<operation name="ProcessNonRMFaultPart">
<participant name="InlineInvokeSubProcessService"/>
<output message="InvokeSubProcessServiceSOATypeInputMessage">
<assign to="WFD_NAME">WS_RequestHandler</assign>
<assign to="NOTIFY_PARENT_ON_ERROR">SERVICE_ERRORS_ONLY</assign>
    <assign to="." from="*"></assign>
</output>
<input message="inmsg">
<assign to="." from="*"></assign>
</input>
</operation>
    </choice>
</sequence>
<operation name="ProcessNonRMPart">
<participant name="InlineInvokeSubProcessService"/>
<output message="InvokeSubProcessServiceSOATypeInputMessage">
<assign to="WFD_NAME">WS_RequestHandler</assign>
    <assign to="NOTIFY_PARENT_ON_ERROR">SERVICE_ERRORS_ONLY</assign>
<assign to="." from="*"></assign>
<input message="inmsg">
    <assign to="." from="*"></assign>
</input>
</operation>
</choice>
<onFault>
    <sequence name="Sequence Start">
        <operation name="SOAFaultService">
            <participant name="SOAFaultService_Instance"/>
            <output message="SOAFaultServiceInputMessage">
                <assign to="faultCode">Receiver</assign>
                <assign to="faultString">Error Processing Request</assign>
                <assign to="faultType">SOAP</assign>
                <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
                <assign to="." from="*"></assign>
            </input>
        </operation>
        <operation name="HTTP Respond Service">
            <participant name="HttpRespond"/>
            <output message="HttpRespondServiceInputMessage">
                <assign to="." from="*"></assign>
            <assign to="doc-has-headers">false</assign>

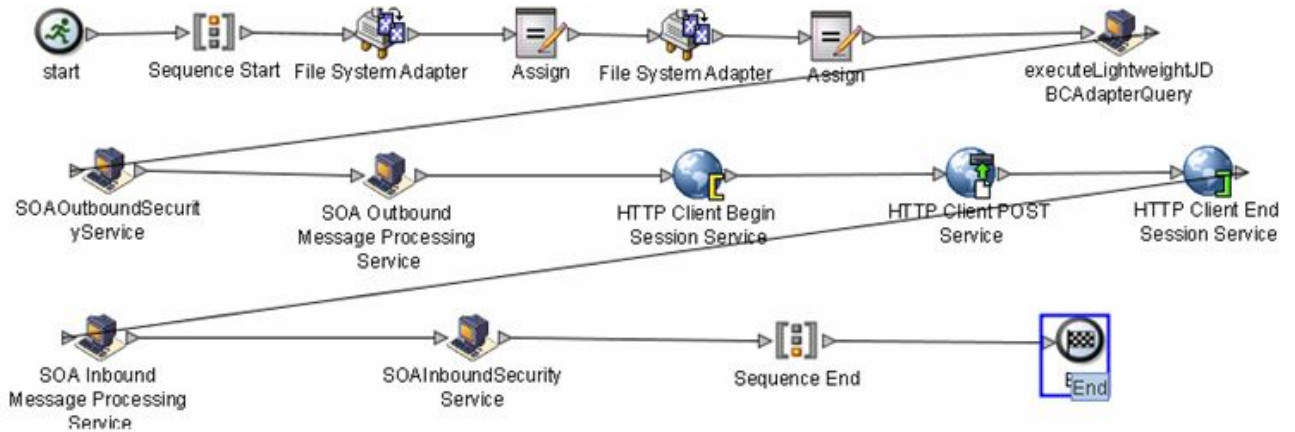
```

```

</output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
</sequence>
</onFault>
</sequence>
</process>

```

Example Business Process as Web Service 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
Attachment ID	Unique ID (MIME content ID) of the attachment, if attachment is passed with the SOAP message
Primary Document	Primary Document
SOAP Document Part	SOAP Document part of the SOAP message

SOA Inbound Security Service

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 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 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 coming to the application (either request or response) is a secure message, this service will be used to understand and process the message to make it compatible with the succeeding nodes/services for further processing.
Usage example	On a consumer side of Web services, the response received is a secure SOAP message. In that case, this service will be used to understand that secure response which includes decryption, signature validation and Security Token validation. On the provider side of, the request received is a secure SOAP message. In that case, this service will be used to understand that secure request which includes decryption, signature validation and Security Token validation.
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 will be used by the SOA Inbound Message Processing Service to process the security header part of an Incoming SOAP Message.
Application requirements	Private Certificate for decryption, Public Certificate for Signature Verification, and Security Token for token validation are required. The document passed to this service must be a proper SOAP document.
Initiates business processes?	No
Invocation	This service is invoked from a business process in case of Web service consumer (must be placed after SOA Inbound Message Processing service). 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.
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	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 wrong certificate configurations.
Restrictions	There are no requirements on the usage of this service. The understanding about the security between a Web service provider and consumer will be offline and this service will come into picture 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 can be 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 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	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. <hr/> Note: See <i>Managing Services and Adapters</i> .

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 what the possible values mean, please refer to the WS-Security specification:

Parameter	Description
Name	Name that the service will have in the application
Description	Description of service
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.)

Parameter	Description
	<ul style="list-style-type: none"> Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list.

Parameters Provided in Request Security Settings (Provider)

Parameter	Description
Decryption Certificate	The name of private certificate that will be used to decrypt the SOAP message
UserName Tokens	The name of security token checked in to the application that will be used for validating the token in the incoming SOAP message
Verification Certificate	The name of public certificate that will be 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	The name of private certificate which will be used to decrypt the SOAP message (in BPML, it will be certificate ID)
SECURITY_TOKEN_NAME	The name of security token checked in the application which will be used for validating the token in the incoming SOAP message
VERIFICATION_KEY	The name of public certificate which will be used to verify the signed SOAP request. (in BPML, it will be certificate ID)

Parameters Passed From Service to Business Process

The output will be a Soap Message with the security header processed and the decrypted body in case the original SOAP message was an encrypted one.

Business Process Example

Application consumer:

```
<process name="WS_BP2">
  <sequence name="First Seq">
    <operation name="File System Adapter">
      <participant name="WS_FSA1"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="attachFile">/ais_local/share/henry/collect/helloWorld.txt
        </assign>
        <assign to="." from="*"></assign>
      </output>
    </operation>
  </sequence>
</process>
```

```

    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
<assign name="Assign" to="HelloWorld_mod"
  from="//PrimaryDocument/@SCIOBJECTID"></assign>
<operation name="File System Adapter">
  <participant name="WS_FSA1"/>
  <output message="FileSystemInputMessage">
    <assign to="Action">FS_EXTRACT</assign>
    <assign to="attachFile">/ais_local/share/henry/collect/mesaAuth.txt</assign>

    <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
<assign name="Assign" to="mesaAuth" from="//PrimaryDocument/@SCIOBJECTID"></assign>

<operation name="executeHelloWorld_mod">
  <participant name="DS_TEST_WS4GSICLIENT_DEC_PORT1_OPE1_Instance"/>
  <output message="DS_TEST_WS4GSICLIENT_DEC_PORT1_OPE1InputMessage">
    <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">henry-edwin</assign>
    <assign to="RemotePort">38240</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="ResponseTimeout">300</assign>
    <assign to="URI">/soap-new?service=WS4GISClient </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="SOAOutboundSecurity_decl"/>
  <output message="SOAInboundSecurityServiceInputMessage">
    <assign to="DECRYPTION_KEY">henry:12:10.14.15</assign>
    <assign to="VERIFICATION_KEY">henry:45.66.99</assign>
    <assign to="." from="*"></assign>
  </output>
<input message="inmsg">
  <assign to="." from="*"></assign>
</input>
</operation>
</sequence>
</process>

```


SOA Outbound Message Processing Service

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 or 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 or 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 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	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 picture if the SOAP message is sent/received by the application.
Persistence level	System default
Testing considerations	Debug information can be found by observing the execution flow of the business process in the application user interface. The level of logging can be controlled through the log.properties file, just as for other log files. Also see Business Usage, Usage Example and Application Requirements.

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. <hr/> <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 Web Service 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.

```
<!-- New BP for the Web Service Layer -->
<process name = "WS_RequestHandler">
  <rule name="ServiceBranch">
    <condition>serviceMode=1</condition>
  </rule>
  <rule name="BPBranch">
    <condition>serviceMode=0</condition>
  </rule>

  <sequence name="ProcessNonRMPart">
    <!-- New SOA Inbound Security Service Handle signature
    verification/decryption part -->
    <operation name="SOAInboundSecurityService">
      <participant name="SOAInboundSecurityService_Instance"/>
      <output message="SOAInboundSecurityServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
```

```

<!-- Updated SOA Request Handler -->
  <operation name="SOARequestHandler">
    <participant name="SOARequestHandler_Instance"/>
    <output message="SOARequestHandlerInputMessage">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

  <choice name="Choice Start">
    <select>
      <case ref="ServiceBranch" activity="DynamicServiceInvoker"/>
      <case ref="BPBranch" activity="Invoke Sub-Process"/>
    </select>
    <operation name="DynamicServiceInvoker">
      <participant name="SOAServiceInvoker_Instance"/>
      <output message="DynamicServiceInvokerInputMessage">
        <assign to="SVC_NAME" from="string(/ProcessData/typeName)"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="Invoke Sub-Process">
      <participant name="InlineInvokeSubProcessService"/>
      <output message="InvokeSubProcessServiceTypeInputMessage">
        <assign to="WFD_NAME" from="string(/ProcessData/typeName)"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </choice>
  <operation name="SOAResponseBuilder">
    <participant name="SOAResponseBuilder_Instance"/>
    <output message="SOAResponseBuilderInputMessage">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
  <operation name="SOAOutboundSecurityService">
    <participant name="SOAOutboundSecurityService_Instance"/>
    <output message="SOAOutboundSecurityServiceInputMessage">
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>

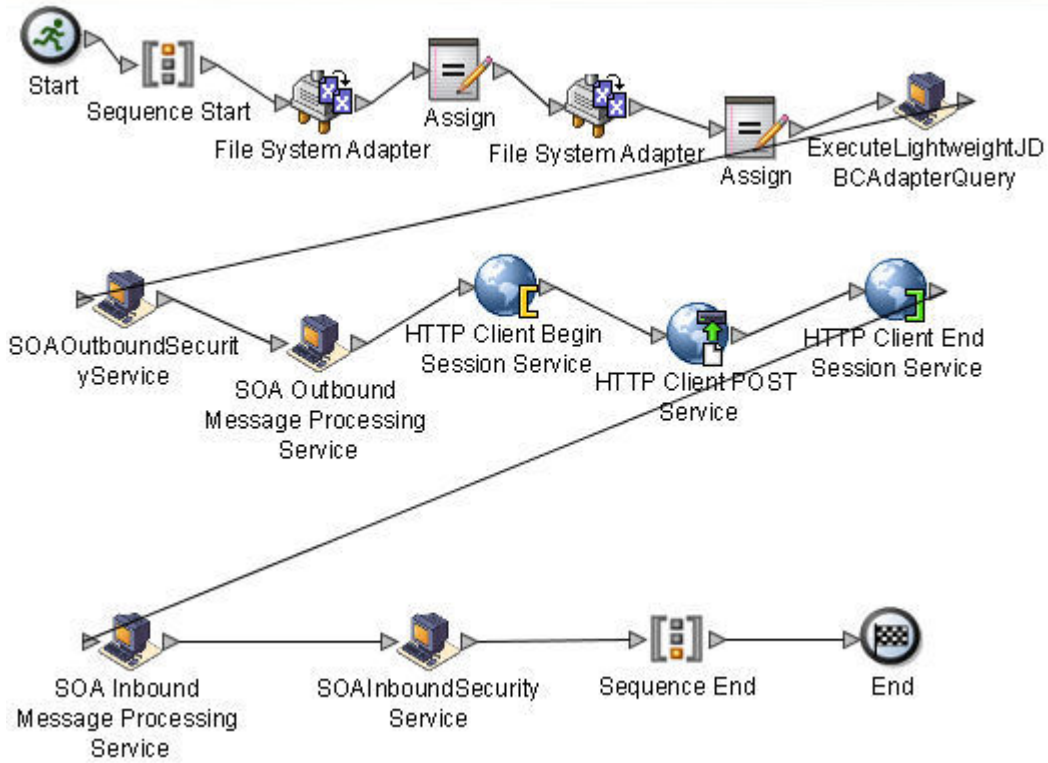
```

```

    </operation>
    <!-- SOAOutbound Service is required to add SOAPAction Header in the
         Primary Document -->
    <operation name="SOAOutboundMsgProcessingService">
<participant name="SOAOutboundMsgProcessingService_Instance"/>
<output message="SOAOutboundInputMessage">
<assign to="." from="*"></assign>
<assign to="ADD_SOAP_ENVELOPING">false</assign>
</output>
<input message="inmsg">
<assign to="." from="*"></assign>
</input>
    </operation>
    <operation name="HTTP Respond Service">
    <participant name="HttpRespond"/>
    <output message="HttpRespondServiceInputMessage">
<assign to="." from="*"></assign>
<assign to="doc-has-headers">>true</assign>
    </output>
    <input message="inmessage">
<assign to="." from="*"></assign>
    </input>
    </operation>
    <onFault>
    <sequence name="Sequence Start">
    <operation name="SOAFaultService">
    <participant name="SOAFaultService_Instance"/>
    <output message="SOAFaultServiceInputMessage">
    <assign to="faultCode">Receiver</assign>
    <assign to="faultString">Error Processing Request</assign>
    <assign to="faultType">SOAP</assign>
    <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
    <assign to="." from="*"></assign>
    </input>
    </operation>
    <operation name="HTTP Respond Service">
    <participant name="HttpRespond"/>
    <output message="HttpRespondServiceInputMessage">
    <assign to="." from="*"></assign>
    <assign to="doc-has-headers">>false</assign>
    </output>
    <input message="inmsg">
    <assign to="." from="*"></assign>
    </input>
    </operation>
    </sequence>
    </onFault>
</sequence>
</process>

```

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 attachment is passed with the SOAP message
Primary Document	Primary Document
SOAP Document Part	SOAP Document part of the SOAP message

SOA Outbound Security Service

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, and other features 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 the application. The obtained certificates need to be checked into the 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	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.

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	Select one of the options: <ul style="list-style-type: none"> • None – Do not include the configuration in a service group at this time.

Field	Description
	<ul style="list-style-type: none"> • 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	This is the 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	This is the signature algorithm to be used for signing the SOAP message.
SIGNING_KEY_IDENTIFIER_TYPE	The identifier type to refer the signing key as, in the signature info in the wsse:Security header.
CANONICALIZATION_ALGO	The canonicalization (formatting) to be applied to the SOAP message before signing it so that signature value is interoperable.
ENCRYPTION_CERTIFICATE	The 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	The Symmetric Key Algorithm used to encrypt the SOAP message.
ENCRYPT_KEY_IDENTIFIER_TYPE	The identifier type to refer the encryption key as, in the wsse:Security header.

Parameter	Description
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	The user name token to be used for the security header. For this drop-down to be populated, at least one Security token must be configured for Web services (Deployment->Web Services->Security Token).
INSERT_TIME_STAMPS	Specified whether a timestamp security header needs to be inserted.
TIME_TO_LIVE	The 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.
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 configured in the application itself (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.

```
<process name="Dynamic_Service_SOAP_BP">
  <sequence>
    <operation name="File System Adapter">
      <participant name="test_fsa"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="attachFile">soapMessage.txt</assign>
        <assign to="deleteAfterCollect">>false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

```

<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>

```

```

</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	SOAP is an XML-based protocol that allows access to services, objects, and servers in a platform-independent manner. 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.
Business usage	Processes inbound SOAP messages.
Usage example	Examples: <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	Related services: <ul style="list-style-type: none">• SOAP Outbound service

	<ul style="list-style-type: none"> • HTTP Server adapter • HTTP Client adapter <hr/> <p>Note: The HTTP Server adapter and HTTP Client adapter replace the B2B HTTP Server adapter and HTTP Send service, which have entered the retirement process. For more information, see Services and Adapters in the Retiring Category.</p>
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	<p>Returned status values:</p> <ul style="list-style-type: none"> • 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

To implement 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, which have entered the retirement process. For more information, see *Services and Adapters in the Retiring Category*.

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.

Field	Description
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 what was specified in the SOAP message. For details, 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 B2B HTTP Server adapter so that it:
 - 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	SOAEnvNSPrefix	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	SOAEnvNSURI	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>
  </choice>
  <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>

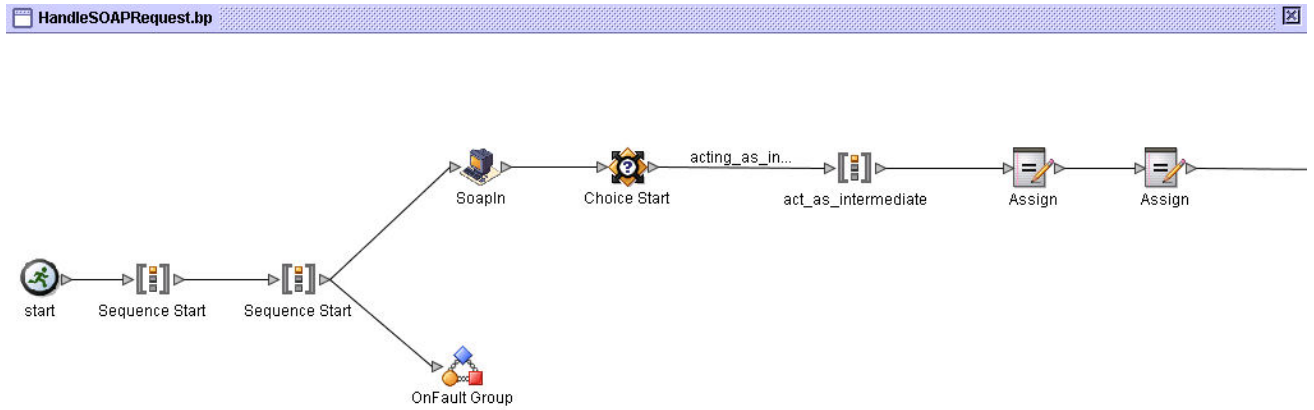
<operation name="SoapOut">
  <participant name="SOAPOutbound"/>
  <output message="output">
    <assign to="." from="*"></assign>
    <assign to="SOAP_MODE">respond</assign>
  </output>
  <input message="input">
    <assign to="." from="*"></assign>
  </input>
</operation>
<assign to="doc-has-headers">true</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>

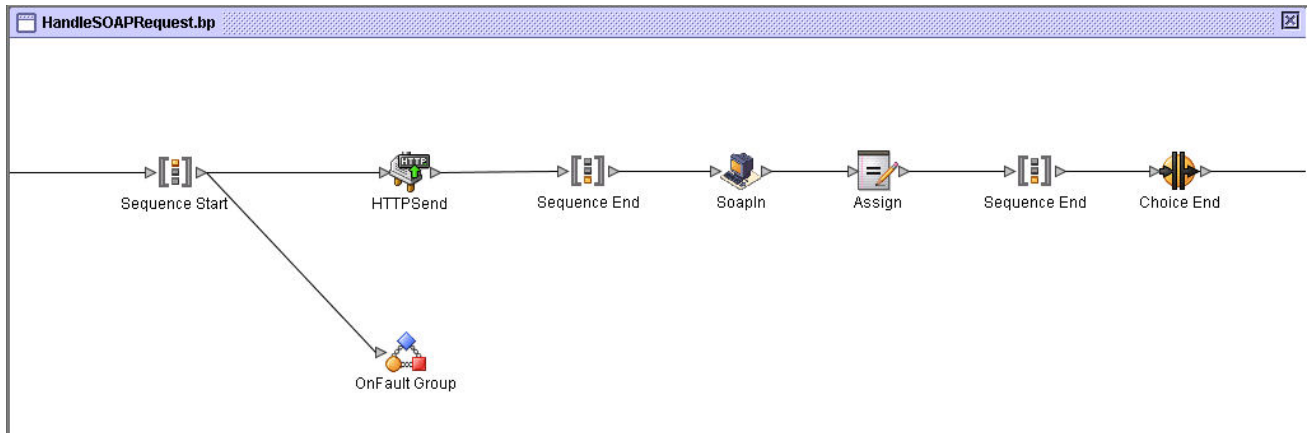
```

The following illustrations show the business process as it appears in the GPM:

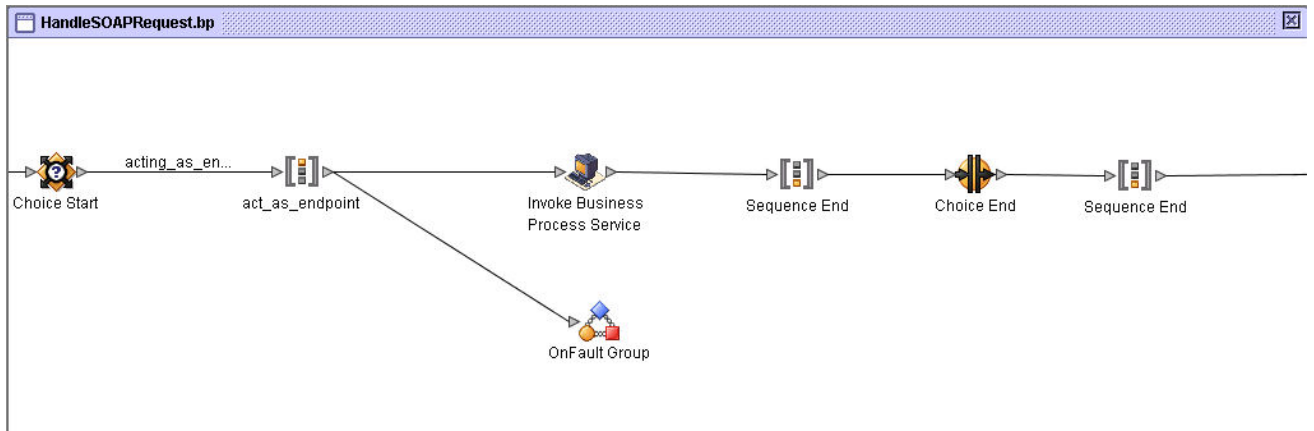
HandleSOAPRequest Screen 1 of 4



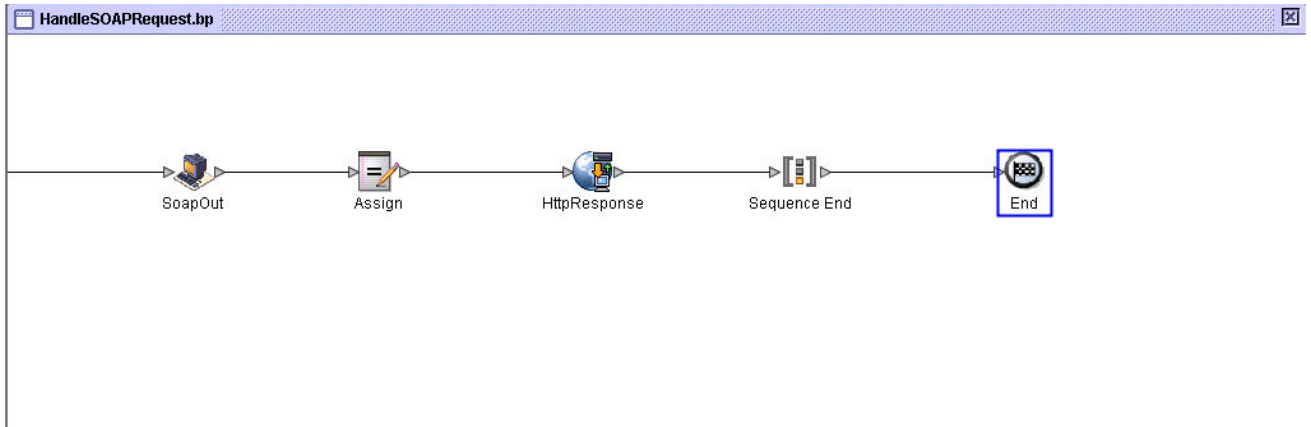
HandleSOAPRequest Screen 2 of 4



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.

Required SOAP Function	SOAP Parameter	Parameter Value
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:

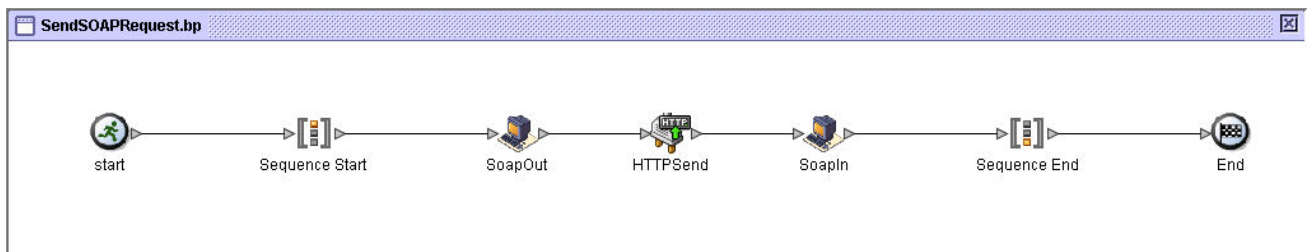
```
<process name="SendSOAPRequest">
  <sequence>
    <operation name="SoapOut">
      <participant name="SOAPOutbound"/>
      <output message="BPML:output">
        <assign to="." from="*"></assign>
      </output>
      <input message="input">
        <assign to="." from="*"></assign>
      </input>
    </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:



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'
      SCIOBJECTID='neith:4aedaa:eb9e01efbe:-7d3e' />
    <SOAPAttachment2 Content-ID='cid:http://productheaven.com/3874923874.xml'
      Content-Location='http://productheaven.com/3874923874.xml'
      SCIOBJECTID='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 SCIOBJECTID='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	SOAP is an XML-based protocol that enablesthe application to access services, objects, and servers in a platform-independent manner. 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 an HTTP Client adapter instance sends the document to the necessary location.
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	<p>Related services:</p> <ul style="list-style-type: none"> • SOAP Inbound service • HTTP Server adapter • HTTP Client adapter <hr/> <p>Note: The 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>Monitoring Deprecated (Retired) Resources</i>.</p>
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 <code>respond</code> . The default value for this parameter is <code>send</code> .
Business process context considerations	None
Returned status values	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, <code>SendSOAPRequest</code> , 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 <code>SOAPEnvAttribute</code> . The value of this parameter contains the complete attribute specification (of the form <code>attributeName="attributeValue"</code>). You can use this parameter multiple times with <code>append="true"</code> 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 <code>"SOAPAddXMLDeclaration"</code> to <code>true</code> or <code>yes</code> .

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 [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

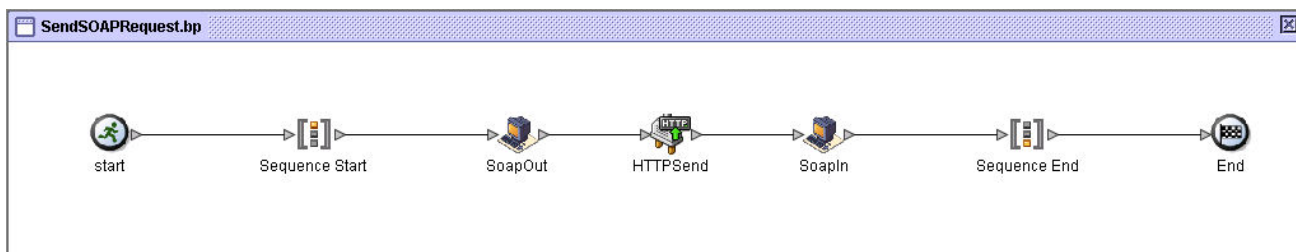
Required SOAP Function	SOAP Parameter	Parameter Value
		if you expect a signature on the SOAP response.
Must use a proxy server	xport-tp-proxyauth	The following information in this format: <i>IP address,port number,username,password.</i> User name and password may not be required.
Envelope namespace 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 Pro-cessData 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/myattachment1.txt</assign>
```

Sending SOAP Fault Messages

The Web Services Interoperability Organization (WS-I) requires all SOAP faults be sent with a 500 HTTP error status. Sterling Integrator sends SOAP faults with a 200 HTTP error status.

Note: If you force a 500 HTTP error status to be returned, a default Jetty error message is returned instead of the SOAP fault document.

To enable Sterling Integrator to send SOAP faults with 500 HTTP error status, add the following assignment to all the SOAP Outbound services used in the business process workflow:

```
<assign to=""></assign>
```

To communicate automatically the parameters with the HTTP Client adapter, direct access to the process data must be provided in the 'MessageFromService' configuration of SOAP Outbound service. The HTTP Client adapter must be able to access the process data in the 'MessageToService' configuration of SOAP Outbound service to enable the HTTP Client adapter to read the parameters passed from the SOAP Outbound service.

If direct access is not provided to the HTTP Client adapter, you must manually pass the following parameters to the HTTP Client adapter:

```
<doc-has-headers>true</doc-has-headers>
<status-code>500</status-code>
<xport-raw-xresponse>true</assign>
<xport-doc-on-error>true</assign>
```

SSH Key Grabber Adapter

The SSH Key Grabber adapter is used by the Known Host Key UI wizard to get the SSH Known Host Key from a trading partner's SFTP or SCP server by way of perimeter services. This adapter is not used as a standalone adapter.

The following table provides an overview of the SSH Key Grabber adapter:

System name	SSH Key Grabber Adapter
Graphical Process Modeler (GPM) category	All Services
Description	This adapter is used by the Known Host Key UI wizard to get the SSH Known Host Key from a trading partner's SFTP or SCP server by way of perimeter services. The SSH Key Grabber Adapter is not used as a standalone adapter.
Business usage	Use this adapter to obtain the SSH key (Known Host Key) from the trading partner's SFTP/SCP servers. By providing the host and port information to the adapter, it will obtain the Known Host Key directly from the server. The SSH key fetched by the adapter can be saved to local disk, checked in to the application database as a Known Host Key, or both. This is a quicker and more convenient option than receiving SSH keys by email (or similar means) from the trading partner and checking them into the application.
Usage example	<p>This adapter is used by the Known Host Key UI wizard to get the SSH key of the trading partner server.</p> <ul style="list-style-type: none"> • The application user provides the URL details (remote host name and the Remote Port) of the remote SFTP/SCP server. • The application user can specify the perimeter server, proxy details, and other connection parameters by editing the default provided service instance of this adapter. • The application uses the SSH Key Grabber Adapter to fetch the SSH key information, and displays it on the UI. If the adapter cannot connect to the remote host, it returns an appropriate error.
Preconfigured?	This adapter must be installed and deployed before it can be invoked.

Requires third party files?	SSH 3SP libraries (currently available with the system)
Platform availability	All supported platforms
Related services	None
Application requirements	When this adapter is configured with a perimeter server set to a mode other than local, 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. Refer to the <i>Perimeter Server</i> documentation for details.
Initiates business processes?	No
Invocation	<p>Invoked by:</p> <ul style="list-style-type: none"> • A default business process is provided with the application, which invokes the SSH Key Grabber adapter. • A default service configuration is also provided for SSH Key Grabber adapter. • In case of changes in the parameters, the default service configuration can be edited to accommodate the changes.
Business process context considerations	The configuration parameters and the outgoing document are picked up by the SSH Key Grabber adapter in the workflow context.
Workflow Context Considerations	None
Returned status values	<p>Values:</p> <ul style="list-style-type: none"> • 0 – Success • 1 – Error
Restrictions	None
Persistence level	System default
Testing considerations	Debug information for this adapter can be found in the SSH Key Grabber Adapter and services log files logs.

Implement the SSH Key Grabber Adapter

To implement the SSH Key Grabber Adapter, complete the following tasks:

1. Configure the perimeter server.
2. Edit the SSH Key Grabber adapter's service instance to configure the perimeter server and proxy server.

Configure the SSH Key Grabber Adapter

To configure the SSH Key Grabber Adapter, edit the out of the box SSH Key Grabber Adapter service instance.

Parameters Passed from Business Process to Service

The following table describes the SSH Key Grabber Adapter parameters passed from the Business Process to the Service:

Field or Parameter Name	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration. Required.
Select a Group	Whether to include this adapter instance in a group. Optional. 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 is then created along with this configuration. • Select Group – If you have already created one or more groups for this service type, select from the list. For more information, see <i>Using Service Groups</i> .
Perimeter Server	Select the perimeter server to use with this adapter. Required. Valid values are: <ul style="list-style-type: none"> • Node1 & local (Default) • Local • External
Local Port Range	Ports that the client can use to establish an SSH channel to the remote server. Optional. Port value can be a single number or a range, list, or a combination.
Connection Retries (ConnectionRetries)	Number of times that the adapter attempts to connect to the remote SSH host. Required. Valid values are: <ul style="list-style-type: none"> • 0 - 50 • Default is 1
Retry delay (ms) (RetryDelay)	Number of milliseconds that the adapter waits between retry attempts. Required. Valid values are: <ul style="list-style-type: none"> • 1- 7200000 milliseconds (7200 seconds) • Default is 5000 ms (5 seconds)
ResponseTimeout (ResponseTimeOut) (secs)	The maximum number of seconds that the trading partner's system can take to respond before the session times out and terminates. Optional. Default value is 40 seconds.
Use Proxy Server (myProxyServer)	User can specify proxy setting for the adapter. Required. Valid values are: <ul style="list-style-type: none"> • Yes • No (Default)

Proxy Configuration Parameters Passed from Business Process to Service

The following table describes the SSH Key Grabber Adapter proxy configuration parameters passed from the Business Process to the Service:

Parm Name (Display Name)	Description
Proxy Host (ProxyHost)	Host name or IP address of the target proxy server. Required. Valid values: <ul style="list-style-type: none"> • Any valid proxy host name or IP address • No default <hr/> Note: Trading Partner profile setting can override this value.
Proxy Port (ProxyPort)	Port number of the target proxy server. Required if Use Proxy Server is Yes. Valid value is any available port. <hr/> Note: Trading Partner profile setting can override this value.
Proxy retries (ProxyRetries)	Number of times the adapter attempts to connect to the proxy server. Required if Use Proxy Server is set to Yes. Valid values are: <ul style="list-style-type: none"> • 0 - 50 • Default 3 This value overrides the Connection retries value.
Proxy Authentication (ProxyBasicAuthentication)	Indicates whether the proxy server requires basic authentication. Required. Valid values are: <ul style="list-style-type: none"> • Yes • No (Default)
UserName (ProxyUserId)	User ID used for proxy authentication. Required if Proxy Authentication is Yes.
Password (ProxyPasswd)	Password used for proxy authentication. Required if Proxy Authentication is Yes.

Proxy Configuration Parameters Passed from Service to Business Process

The following table describes the SSH Key Grabber Adapter proxy configuration parameters passed from the Service to the Business Process:

Parameter	Description
Remote Host (rHost)	External Trading Partner host system (SFTP Server IP Address or DNS name). Required. Valid values are: <ul style="list-style-type: none"> • Valid IP address • DNS name
Remote Port (rPort)	External Trading Partner port number. Valid value is any valid port number. No default value.

Parameter Passed from Service to Business Process

The following table describes the SSH Key Grabber Adapter parameter passed from the Service to the Business Process:

Parameter Name	Description
Primary Document	SSH key information fetched by adapter and placed in the primary document.

SSLCertGrabber Adapter

The following table provides an overview of the SSLCertGrabber adapter:

System name	SSLCertGrabber Adapter
Graphical Process Modeler (GPM) category	All Services
Description	This adapter captures SSL certificates of trading partner servers supporting HTTPS/FTPS protocols.
Business usage	<p>This adapter is used by Cert Capture Utility Wizard to capture the certificates of the trading partner server with which future transactions might happen. Once the certificate(s) of the HTTPS/FTPS server is obtained, they can be manually verified and then checked in to the System as Trusted/CA Certificate based on the certificate type.</p> <p>The Adapter parameters hostname and port name are taken from the UI Wizard.</p>
Usage example	Default service configuration should be edited by the user to provide the connection parameters like proxy information(if required) and Perimeter Server information. The UI Wizard obtains the host name and port of trading partner server and invokes a Sterling Integrator Business Process which uses the adapter to fetch trading partner certificates.
Preconfigured?	The SSLCertGrabber Adapter must be installed and deployed before it can be used.
Requires third party files?	Certicom SSL Library (currently available with Sterling Integrator)
Platform availability	All supported platforms
Related services	None
Application requirements	<p>An HTTP/FTP server at the external trading partner location with SSL enabled.</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 SI by a firewall. Please</p>

	refer to the Perimeter Services documentation for details on installing and running that component.
Initiates business processes?	No
Invocation	<ol style="list-style-type: none"> 1. A System BP is provided with , which invokes the SSLCertGrabberAdapter 2. A service instance with default configuration is also provided with Sterling Integrator for this adapter 3. In case of changes in the parameters, the system provided service configuration can be edited to accommodate the changes.
Business process context considerations	None
Returned status values	<p>The possible status values that are returned from this service.</p> <ul style="list-style-type: none"> • 0 - Success • 1 - Error
Restrictions	None
Persistence level	Default
Testing considerations	Debug information for this adapter can be found in the SSLCertGrabberAdapter and Services log files.

Implementing the SSLCertGrabber Adapter

To implement the SSLCertGrabber adapter, complete the following tasks:

1. Create an SSLCertGrabber adapter configuration. See *Managing Services and Adapters*.
2. Configure the SSLCertGrabber adapter. See *Configuring the SSLCertGrabber Adapter*.

Configuring the SSLCertGrabber Adapter

To configure the SSLCertGrabber adapter:

1. Select **Deployment > Services > Configuration**.
2. Search for SSLCertGrabber adapter or select it from the list.
3. Click **Edit**.
4. Specify field settings:

Field	Description
Name	Name this adapter will have in
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.

Field	Description
	<ul style="list-style-type: none"> • 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>
Local port range	User can specify a port range which the client can use for established connection. Optional.
Global Proxy Server	Global proxy setting can be specified incase local adapter proxy setting are absent. Optional.
Number of connection retries	User should specify no. of connection retries for the adapter. Required.
Delay between retries	User should specify delay before adapter will retry to establish connection. Required.
Delay Waiting On IO	User should specify the time the adapter will wait for IO while establishing connection. Required.
Use Proxy Server	User can specify proxy setting for the adapter (only valid for HTTPS). Required.
Proxy Hostname	IP address or host name of the target proxy server. Trading Partner profile setting can override the value. Required (if Use proxy server is Yes).
Proxy Port	Port number of the target proxy server. Trading Partner profile setting can override the value. Required (if Use proxy server is Yes).
Proxy Retry Attempts	The number of times the Adapter will try to connect to proxy server. Trading Partner profile setting can override the value. Required (if Use proxy server is Yes).
Use Proxy Basic Authentication	Indicates if the proxy server requires basic authentication. If Yes, a trading partner profile is necessary because user credentials will come from the profile. Required.

Parameters Passed from BP to Service

Field	Description
Host name or IP Address	Trading partner server's domain name or IP address. Required.
Port	Port supporting SSL protocol. Required.
SSL Mode	These modes are available only for FTPS. Required.
Protocol	Protocol supported by trading partner server for SSL connection. Required.

Parameters Passed from Service to BP

Field	Description
Certificate [0-n]	SSL Certificate that the adapter fetched.

Business Process Example of Cert Capture Utility

The following example illustrates Cert Capture Utility.

Using the SSLCertGrabberAdapter to connect to Trading partner FTP Host TP_HOST & TP_PORT, Obtain Certificates of the same server and extract them to local system to manually verify the certificates. Later they can be checked in to SI system to carry on with future transactions with Trading Partner's Server.

Here, A BP is created which invokes the default BP SSLCertGrabberAdapterBP that comes with SI. The Host Address TP_HOST and Port number TP_PORT are provided as input to the BP using assign statements and connection attempt is made by the SSLCertGrabberAdapter, which will fetch the certificates of the business server.

Assumption: TP_HOST & TP_PORT exists and must be SSL Enabled, Also it supports explicit mode.

```
<process name="SSLCertGrabberAndExtractor_FTPS">
  <sequence name="Sequence Start">
    <operation name="Invoke Business Process Service">
      <participant name="InlineInvokeBusinessProcessService"/>
      <output message="InvokeBusinessProcessServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="WFD_NAME">SSLCertGrabberAdapterBP</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
        <assign to="RemoteHost">TP_HOST</assign>
        <assign to="RemotePort">TP_PORT</assign>
        <assign to="SSLMode">FTPS</assign>
        <assign to="Protocol">Explicit</assign>
      </input>
    </operation>
    <assign to="PrimaryDocument" from="Certificate1/@*" append="false"/>
  </sequence>
</process>
```

Note: For security reasons, do not use any actual machine, server, user, or domain names or IDs; URLs, directory paths, IP addresses, passwords, etc., in your samples.

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	Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator. The Sterling Control Center receives the information about the event through a SOAP request to the Sterling Control Center service in Sterling Integrator. 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator Web Services Manager configuration.

Configuring the Sterling Control Center Service

The Sterling Control Center service is configured upon installation of Sterling Integrator. 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	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: <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> .

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	Request type sent from the Sterling Control Center to the Sterling Control Center service. Required. Valid values are: <ul style="list-style-type: none"> • getHistory • getBPHistory • getInfo • getStatus • getLicense • OpsCommand • runSql
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 Sterling Integrator. Required if Request Type = Command. Valid values are: <ul style="list-style-type: none"> • ADAPTERSTATUS • consolidatedstatus • LISTADAPTERS • LISTPERIMETERCLIENTS
Sql	SQL to execute in Sterling Integrator. 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.

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 Sterling Integrator 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 Sterling Integrator.
Requires third party files?	No
Platform availability	All supported Sterling Integrator platforms
Related services	Mapped Extraction service
Application requirements	The fact model referenced by this service must be created and checked in to Sterling Integrator.
Initiates business processes?	Cannot initiate a business process.
Invocation	From a business process.
Business process context considerations	None
Returned status values	Returned status values: <ul style="list-style-type: none"> • 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.
2. 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 Sterling Integrator 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	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	The BP Metadata Info service enables you to retrieve information about a particular business process including: <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	The Get Document Info service provides information about the primary document. You can query the service about information including the following: <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

Service	Function
	<ul style="list-style-type: none"> • Previous Document ID • Document Length
Release Service	The Release service is a system service used to discard previous assignments in process data.
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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • Success • Failure
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 Sterling Integrator 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.

Parameter (BPML Element Value)	Description
Document Identifier (documentId)	The ID assigned by Sterling Integrator 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://www.java.sun.com/j2se/1.3.1/docs/api/index.htm 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 Sterling Integrator 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 Sterling Integrator.
Timestamp (time)	Current timestamp.

Parameter (BPML Element Value)	Description
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.

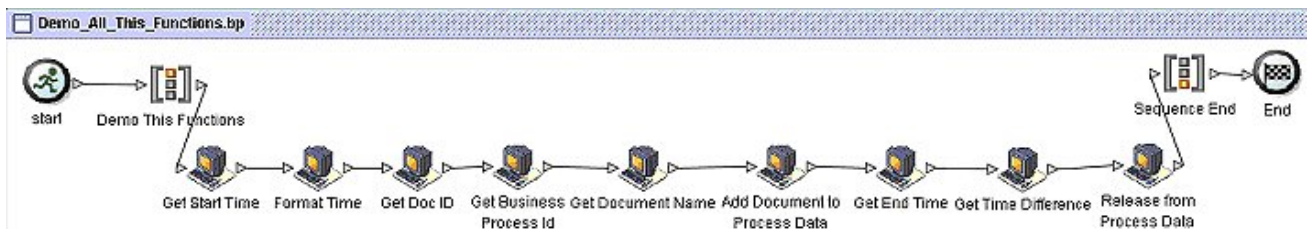
Usage Examples

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**.

The screenshot shows the 'Service Editor-Get Start Time' window. The 'Name' field is 'Get Start Time' and the 'Config.' field is 'This'. The 'Message From Service' tab is selected. The 'Input Msg' dropdown is set to 'Allow message write'. Below this is a table with the following data:

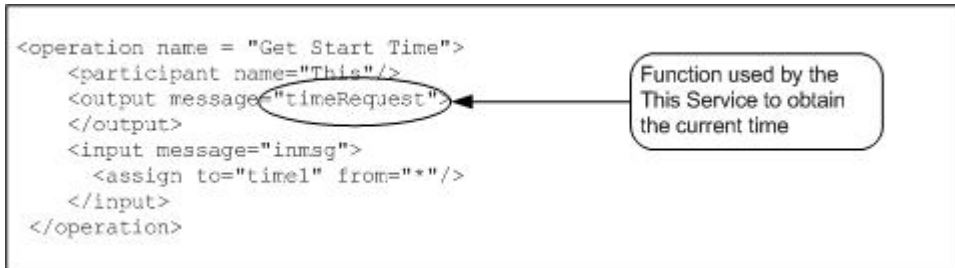
Name	Value	Use XPATH?
time1	*	<input checked="" type="checkbox"/>

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.

The screenshot shows the 'Service Editor-Get Start Time' window. The 'Name' field is 'Get Start Time' and the 'Config.' field is 'This'. The 'Message To Service' tab is selected. The 'Output Msg' dropdown is set to 'Messages Only' and the 'Message Name' field is 'timeRequest'.

The following illustration shows the BPML:



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>

```

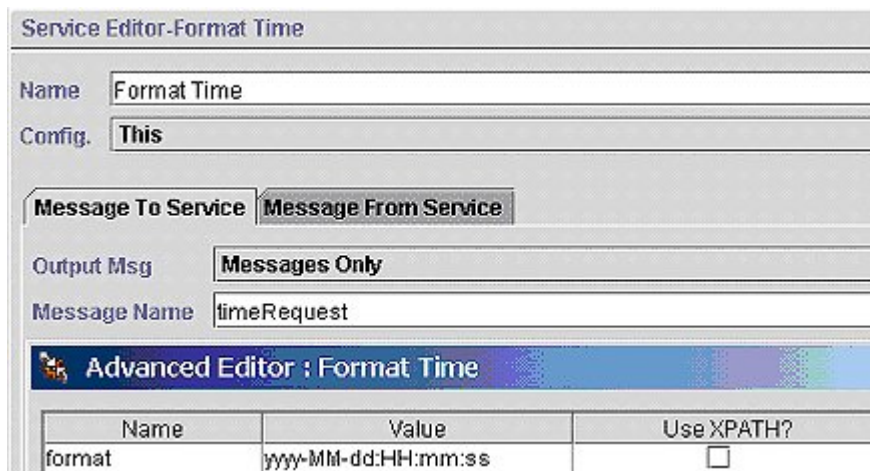
Format Time (timeRequest function)

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://www.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**.



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>

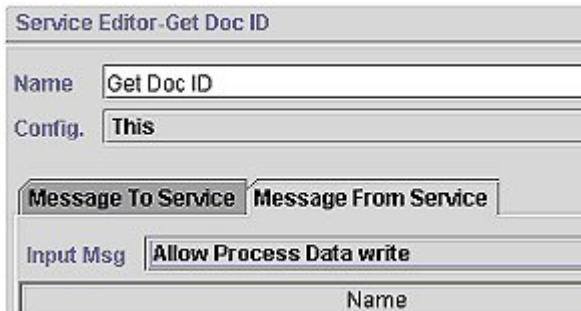
```

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**.

Service Editor-Get Doc ID

Name: Get Doc ID

Config: This

Message To Service | Message From Service

Output Msg: Obtain Process Data first, then Messages

Message Name: giveDocumentRequest

Advanced Editor : Get Doc ID

Name	Value	Use XPath?
documentKey	'PrimaryDocument'	<input checked="" type="checkbox"/>

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:fae7d3336aa:-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**.

Service Editor-Get Business Process Id

Name: Get Business Process Id

Config: This

Message To Service | Message From Service

Input Msg: Allow message write

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`.

Service Editor-Get Business Process Id

Name: Get Business Process Id

Config: This

Message To Service | Message From Service

Output Msg: Messages Only

Message Name: 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**.

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 :

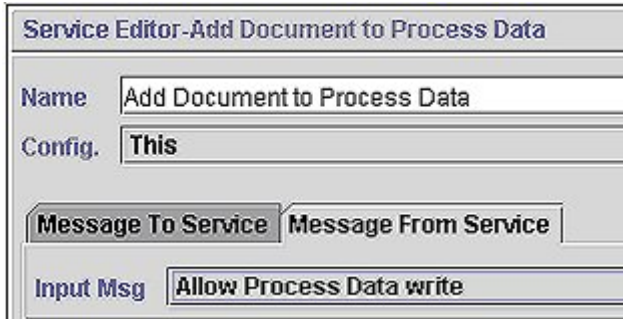
```
<documentName>SampleDocument.xml</documentName>
```

Add Document to Process Data (takeDocumentRequest)

The Add Document to Process Data configuration of the This service adds a document to process data.

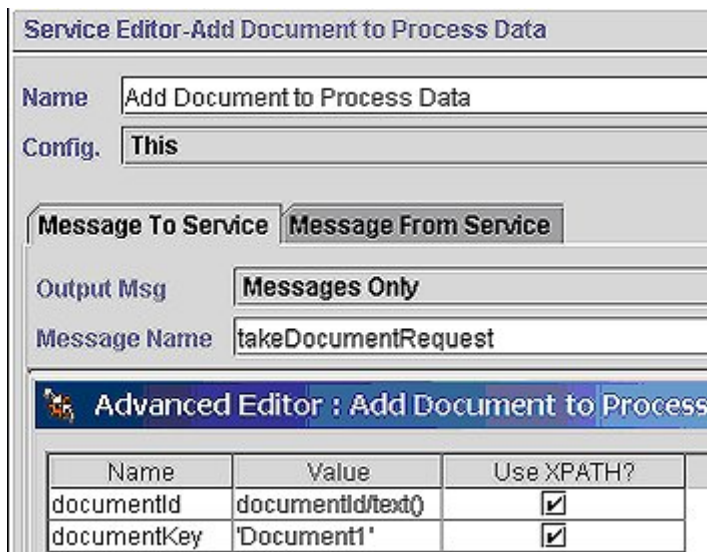
Set the following parameter in the GPM Message From Service tab:

- 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)*.
4. Assign Document1 to **documentKey** in the Advanced Editor, select the **Use XPATH?** check box, and click **OK**.



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:fae7d336aa:-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:

1. Select **Allow message write** from the Input Msg drop-down list and click **Add**.
2. In the Advanced Editor, assign `elapsedTimeMillis` to **diffTimestamp**, select the **Use XPATH?** check box, and click **OK**.

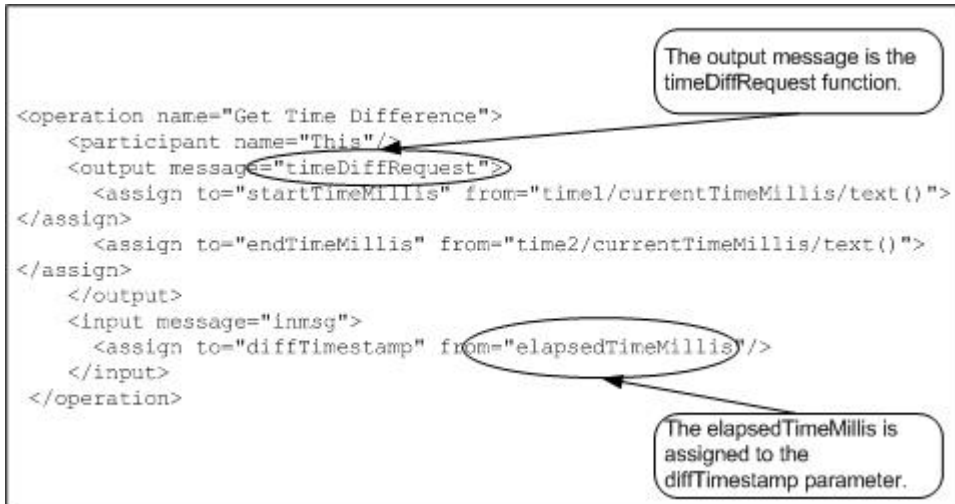
Name	Value	Use XPATH?
diffTimestamp	elapsedTimeMillis	<input checked="" type="checkbox"/>

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 `timeDiffRequest` and click **Add**.
3. 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.
4. Assign `time2/currentTimeMillis/text()` (the end time previously placed in process data) to **endTimeMillis**, select the **Use XPATH?** check box, and click **OK**.

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>

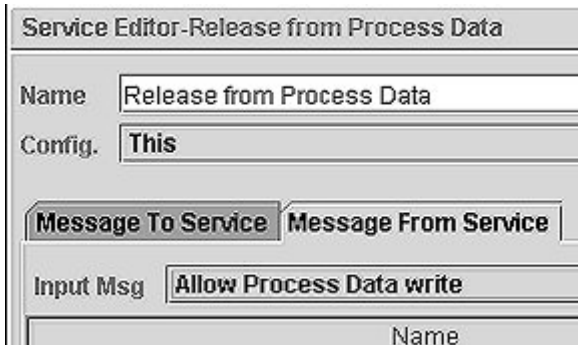
```

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:

```
<ProcessData>
  <PrimaryDocument SCIOBJECTID="L2000-000248:f96c:fae7d336aa:-784a"/>
</ProcessData>
```

Primary document appears in process data.

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 Sterling Integrator 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> .
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 Sterling Integrator.

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*.
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 Sterling Integrator:

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, rvd, with rvd 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 <code>tibrvj.jar</code> file, which contains the TIBCO API class files
LD_LIBRARY_PATH	Include the following library (*.so) files (provided by TIBCO): <ul style="list-style-type: none"> • <code>librv.so</code> • <code>librvcm.so</code> • <code>librvcmq.so</code> • <code>librvkst.so</code> • <code>librvft.so</code> • <code>librvjcms11.so</code> • <code>librvjfts11.so</code> • <code>librvjs11.so</code> • <code>librvts.so</code> • <code>librvXt.so</code> • <code>libtibrv.so</code>

Path	Requirement
	<ul style="list-style-type: none"> • 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.
Daemon	Name of the background process that completes the information pathway between TIBCO Rendezvous processes across the network. The name has one format, rvd, with rvd 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.
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, rvd, with rvd 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 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 Sterling Integrator 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	<p>Invoke the Timestamp Service for PeopleSoft from a business process as shown in the following example:</p> <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	Returned status values: <ul style="list-style-type: none"> • 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.

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 Sterling Integrator 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.

The Timestamp Utility service calculates the time difference using the UNIX time stamp. The time difference is the number of seconds between the particular date and time and the UNIX epoch time.

Note: Although the Timestamp Utility service converts the UNIX time stamp to a calendar or clock time stamp, it does not convert a calendar or clock time stamp to a UNIX time stamp.

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*.
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.

Field	Description
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	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. The Timezone Offset service was created for use with the Oracle E-Business Suite adapter.
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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • Success – The service successfully computed the time zone offset

	<ul style="list-style-type: none"> • 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 Sterling Integrator 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 Sterling Integrator
- Are enabled in Sterling Integrator

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 Sterling Integrator, 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 Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator platforms
Related services	No

How the User Service Works

The User service is used by both Sterling Integrator and by business processes that reference a User service:

- For Sterling Integrator, 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 Sterling Integrator 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 Sterling Integrator 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> .
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: [hostname/ip-address]:ip-port-number/credential-typeexample: 127.0.0.1:6000/tcp:
LD_LIBRARY_PATH	Include the following library (*.so) files (provided by Vitria): <ul style="list-style-type: none">• libiconv3_o3gri.so• libnpjavanative3.so

Path	Requirement
	<ul style="list-style-type: none"> • 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 Sterling Integrator.
 - 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*.
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 Sterling Integrator:

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	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

Field	Description
	<ul style="list-style-type: none"> 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.
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. Sterling Integrator 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 developeped before proceeding.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported Sterling Integrator 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	Returned status values: 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	<p>The Wait Notify service must be used with a custom service that sets the following fields in the service before starting the subprocesses. Trigger:</p> <pre>iwfc.setOrigWFCID(invoking service wfc) iwfc.addContentElement("PARENT_SERVICE_NAME"+ invoking service wfc, wfc.getServiceName) iwfc.addContentElement("PARENT_WF_ID",bootstrapping wf id)</pre> <p>After all subprocesses are started, the service has to call the following:</p> <pre>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());</pre> <p>Note: The Wait Notify service is called by the system automatically if the fields in the table, <i>Output from Business Process to Service</i>, are set before it calls <code>InitialWorkFlowContext(iwfc).start()</code>.</p>
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 Sterling Integrator 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	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	For short wait intervals (less than 30 minutes), Sterling Integrator 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). 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.
Persistence level	Full

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 Sterling Integrator, 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.
UNITS	Minutes, Seconds, Timestamp

webMethods Adapter

The following table provides an overview of the webMethods® adapter:

System name	WebMethodsPublisher, WebMethodsSubscriber
Graphical Process Modeler (GPM) category	None
Description	Connects Sterling Integrator 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> .
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 Sterling Integrator to publish events to the webMethods Enterprise server. For more information about the webMethods Publisher adapter, see *Implementing the webMethods Publisher Adapter*.
- webMethods Subscriber adapter – The webMethods Subscriber adapter enables users of Sterling Integrator 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 Sterling Integrator. For more information about the webMethods Subscriber adapter, see *Implementing the webMethods Subscriber Adapter*.

Basic Operations

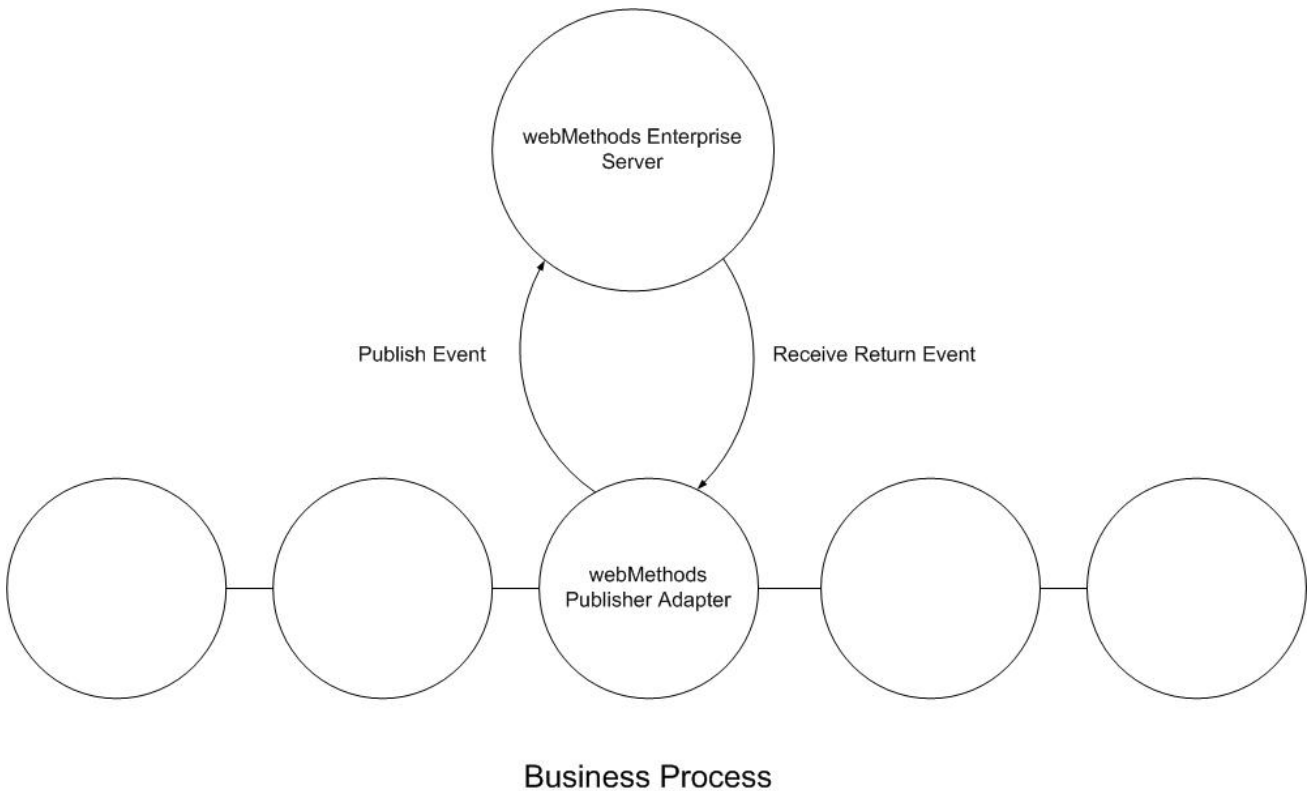
The webMethods adapter supports four basic operations:

- 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.



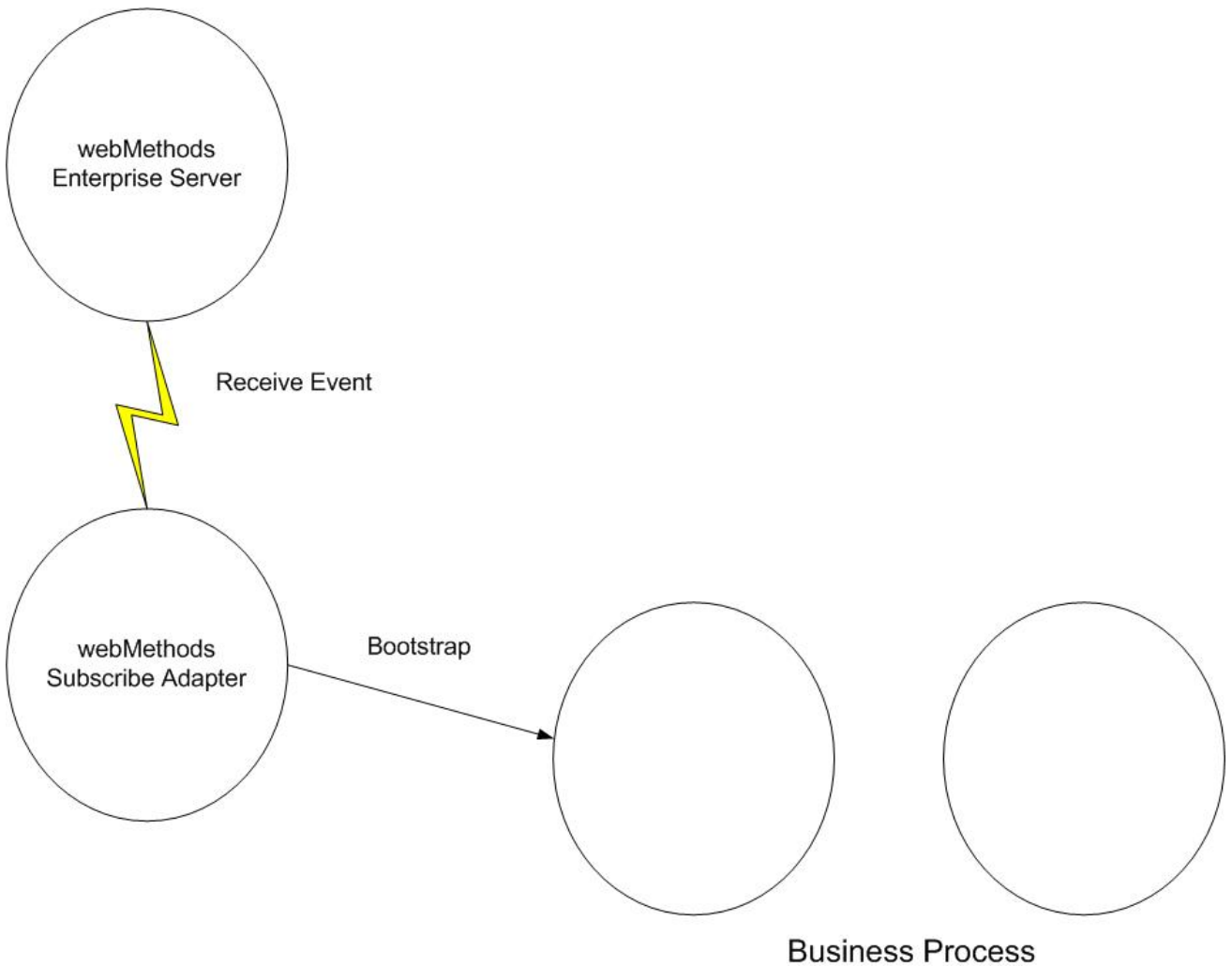
The following steps describe the synchronous request/response operation:

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 Sterling Integrator starts.

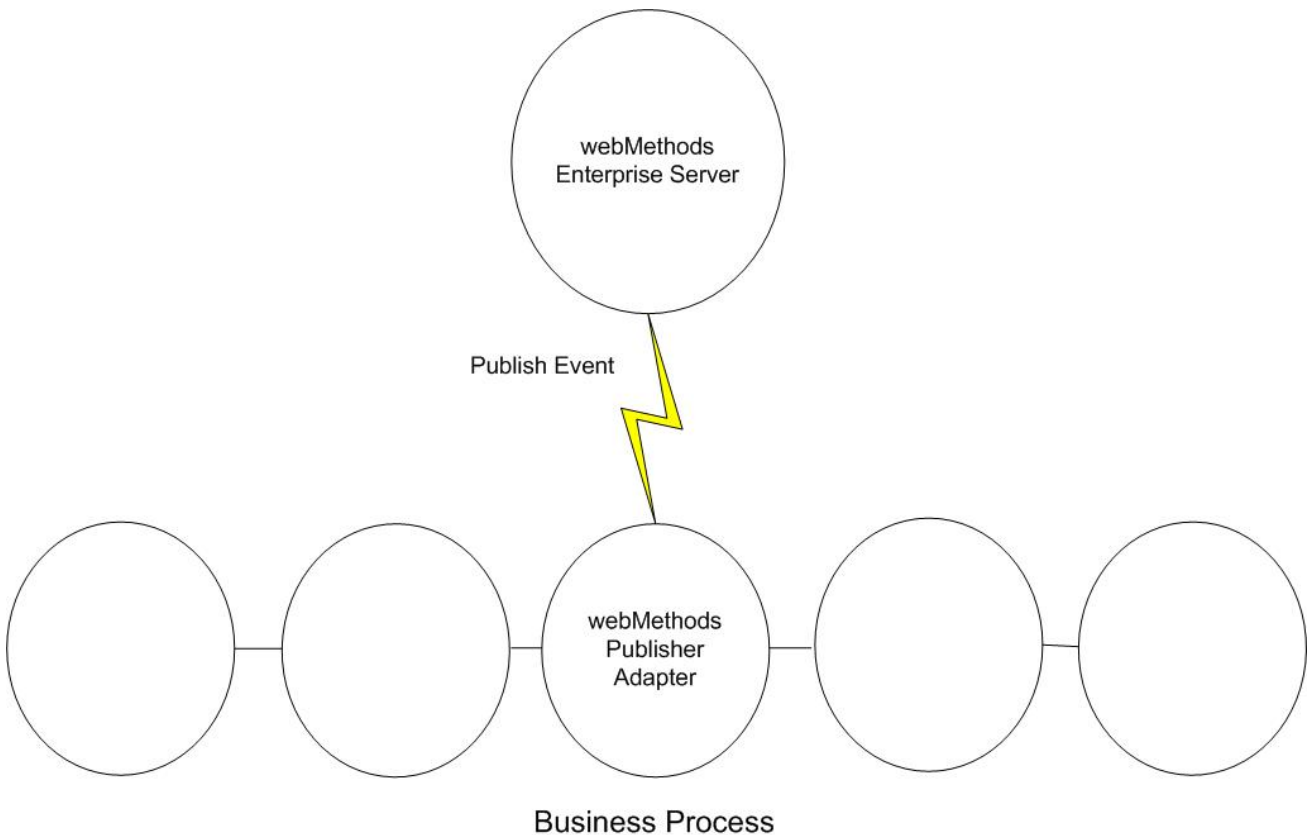


The following steps describe the asynchronous business process start operation:

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.



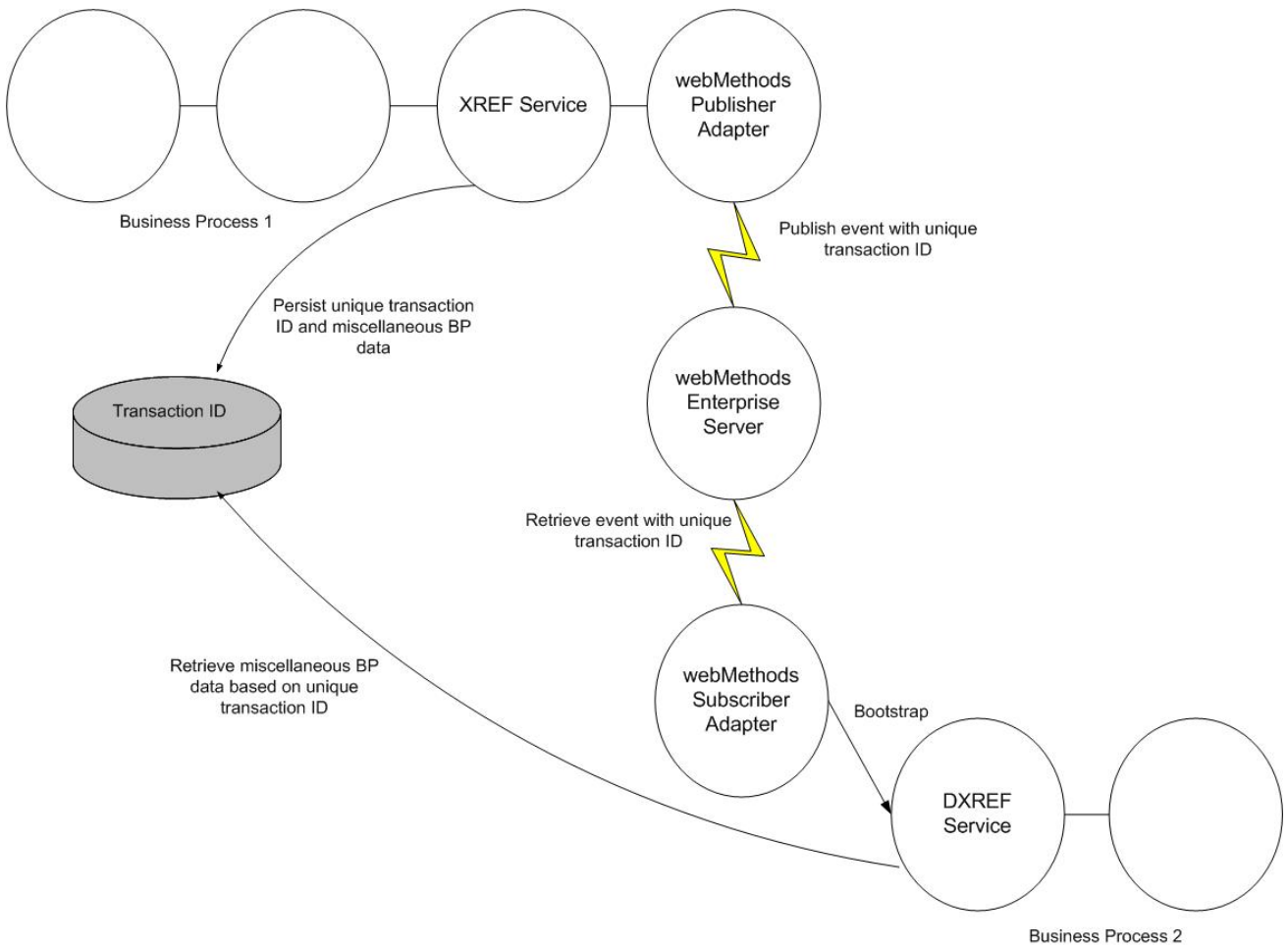
The following steps describe the fire-and-forget operation:

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 Sterling Integrator.
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 Sterling Integrator 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*.

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*.

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*.

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>
</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*.

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 version -l absolutePath/filename
```

For example:

```
./install3rdParty.sh webmethods 5.0 -l /home/webmethods/libawssl50jn.so
```

Implementing the webMethods Publisher Adapter

The webMethods Publisher adapter enables business processes that run in Sterling Integrator 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*.
3. Use the webMethods Publisher adapter in a business process.

Configuring the webMethods Publisher Adapter

Sterling Integrator Configuration

The following table describes the fields used to configure a webMethods Publisher adapter in Sterling Integrator:

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. <hr/> <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
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.
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.

Field	Description
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>
<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 Sterling Integrator. If they are not specified in Sterling Integrator, 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 Sterling Integrator. If they are not specified in Sterling Integrator, 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 Sterling Integrator:

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 Sterling Integrator.
5. Sterling Integrator 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 Sterling Integrator users to subscribe to webMethods events. When an event is received from the webMethods Enterprise server, a user-specified business process is started in Sterling Integrator.

When you create and save a configuration of the webMethods Subscriber adapter, the adapter starts and subscribes to all events specified in Sterling Integrator. No business process can explicitly run this adapter. You must specify a business process related to each event. This is the business process Sterling Integrator 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*.
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 Sterling Integrator and connections for the webMethods Subscriber adapter in a property file.

Sterling Integrator Configuration

The following table describes the fields used to configure the webMethods Subscriber adapter in Sterling Integrator:

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. <hr/> <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
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.

Field	Description
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.
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. 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 Sterling Integrator. • If more than one wildcard is specified for an event, an error occurs when the adapter configuration is saved in Sterling Integrator.

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 Sterling Integrator 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. 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. 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. 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 Sterling Integrator 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 Sterling Integrator.

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 Sterling Integrator.

The adapter runs until one of the following events occurs:

- You stop the webMethods Subscriber adapter from Sterling Integrator.
- You update and save the adapter configuration in Sterling Integrator, 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 Sterling Integrator. 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 Sterling Integrator.
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

Error Message	Description
	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 Sterling Integrator 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.
-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 <code>none</code> 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

Option	Definition
	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* and *Creating a webMethods Subscriber Adapter Configuration*.

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*.

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*.

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. Sterling Integrator 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 WebSphereMQ Suite and the WebSphere adapter now support SSL. For additional information about configuring SSL for WebSphere MQ, see *Configuring the WebSphere MQ 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/java/lib</i> on the WebSphere MQ server. For MQ version 7.0.x and above, all jar files contained in the <i>install_dir/java/lib</i> directory should be installed using the Sterling Integrator <i>install3rdParty</i> utility.</p> <hr/> <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 Sterling Integrator must match the version running on the WebSphere MQ server.</p> <hr/>

Platform availability	All supported Sterling Integrator 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	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 Sterling Integrator is installed.

Note: For more information about the WebSphere MQ Classes for Java, go to the IBM WebSphere MQ site at <http://www-01.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 Sterling Integrator is installed. In the space below, record the path and name of the directory.

Path: _____

3. Shut down Sterling Integrator if it is running.
4. From the bin directory where Sterling Integrator is installed, install the vendor library package (com.ibm.mq.jar) by running the install3rdParty script included with Sterling Integrator. 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 Sterling Integrator.

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 Sterling Integrator is installed. In the space below, record the path and name of the directory.

Path: _____

3. Shut down Sterling Integrator if it is running.
4. From the bin directory where Sterling Integrator is installed, install all the vendor library packages by running the install3rdParty script included with Sterling Integrator. 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:

```
install3rdParty.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 Sterling Integrator.

Configuring the WebSphere MQ Adapter

To configure the WebSphere MQ adapter, you must specify field settings in Sterling Integrator and in the Graphical Process Modeler (GPM). For general information about service and adapter configurations, see *Managing Services and Adapters*.

Sterling Integrator Configuration

The following table describes the fields used to configure the WebSphere MQ adapter in Sterling Integrator:

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	<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. <hr/> <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>
Host Name (mq_hostname)	<p>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.</p> <p>The WebSphere MQ server supports multiple hosts in the Host Name field. The functionality works with MQ 7.0.1 and higher.</p> <p>The Websphere MQ Adapter connects to the first host. If the first host is down, it connects to the next host in sequence until a connection is established. You have to configure multiple MQ hosts with same listening port, Queue Manager, Queue Name, Server Connection Channel to achieve failover.</p> <p>In the host name (mq_hostname) field, specify multiple hosts separated by comma. If mq_action is set to MQ_RCVASYNC, you must specify the values of Failure retry attempts (rcv_retryCount) and Delay between retries (rcv_retrySleep).</p>
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.

Field	Description
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.
CCSID (mq_ccsid)	<p>If needed, enter the Coded Character Set Identifier (CCSID) that represents the codeset name you wish to use. Optional.</p> <p>Codeset name Use CCSID</p> <p>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</p> <hr/> <p>Note: CCSID is not used when connecting directly using “binding moce.”</p>
Minimum Connections (mq_minConx)	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.
Maximum Connections (mq_maxConx)	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.
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) <hr/> <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>
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

Field	Description
	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)	<p>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 yyyymmddhhmsslll.dat. Optional – uses default if not specified in the UI or GPM.</p> <hr/> <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_SYNCPOINT (rcv_MQGMO_syncpoint)	<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> <hr/> <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_FAIL_IF QUIESCING (rcv_MQGMO_failifquiescing)	<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> <hr/> <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_LOGICAL_ORDER (rcv_MQGMO_logicalorder)	<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> <hr/> <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>

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> <hr/> <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> <hr/> <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> <hr/> <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> <hr/> <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> <hr/> <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
Bootstrap Workflow (rcv_workFlowName_bootstrap)	<p>The business process name to start when data is received. Required if using Receive Async, otherwise optional.</p> <hr/> <p>Note: This field is only displayed if this is a Receive Async configuration. For Receive Sync, set this in the GPM.</p>
Async Failure Workflow (rcv_workFlowName_failure)	<p>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.</p>
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> <hr/> <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. 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. 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> <hr/> <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> <hr/> <p>Note: This field is only valid if this is a Receive Async configuration.</p>

Field	Description
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.
Use the group status flag to determine end-of-group? (rcv_useGroupStatus)	<p>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 Receive all messages in a logical group as one document. Valid values:</p> <ul style="list-style-type: none"> • Yes • No
Group messages even when the identifier is MQ*_NONE? (rcv_groupMsgWhenIdNone)	<p>Whether to group messages when the identifier equals MQ*_NONE. Only used if Message Handling is set to Receive all messages in a logical group as one document. Valid values are:</p> <ul style="list-style-type: none"> • Yes • No
User	<p>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.</p>

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 Sterling Integrator 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. 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</p>

Field	Description
	<p>as not setting it at all - the default behavior is used. 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 Sterling Integrator 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> .
rcv_groupBy	Which identifier to use when grouping messages. Only used if rcv_msgHandling is set to Receive all messages in a logical group as one document . 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 rcv_MQMO_corId 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 rcv_MQMO_grpId 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 rcv_MQMO_msgId 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.

Field	Description
rcv_MQMO_grpld	Sets the MQMO_MATCH_GROUP_ID option to this value. Optional.
rcv_MQMO_msgld	Sets the MQMO_MATCH_MSG_ID. Optional.
snd_HexToByte_actToken	Select Yes if the value specified in snd_MQMD_accountingToken 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_correlId	Select Yes if the value specified in snd_MQMD_correlId 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_grpld	Select Yes if the value specified in snd_MQMD_groupId 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_msgld	Select Yes if the value specified in snd_MQMD_msgld 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 - 9999999999.
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.

Field	Description
	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 snd_msgType 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.
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.

Field	Description
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_failIfQuiescing	Used to add the MQPMO_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).
snd_MQPMO_newCorId	If yes, causes the queue manager to generate a new correlation identifier in the CorrelId 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_newMsgId	If yes, causes the queue manager to generate a new message identifier in the MsgId 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).

Field	Description
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.
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).

Field	Description
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.

Configuring SSL Enhancements for WebSphere MQ

The WebSphereMQ Suite and the WebSphere adapter now support SSL with these enhancements:

- SSL cipher specification at channel level for the WebsphereMQ Suite Async Receive adapter
- Parameters for server and client authentication

Configuration to support SSL consists of two steps:

- Configure server for authentication
- Configure client for authentication

Plan for Configuration

The cipher suites and specifications in the following table are supported by IBM WebSphere MQ. However, not all are supported on WebSphere MQ versions below 6.0.

Cipher Suite	Cipher Specification	Supported below MQ v6.0?
SSL_RSA_WITH_NULL_MD5	NULL_MD5	NO
SSL_RSA_WITH_NULL_SHA	NULL_SHA	NO
SSL_RSA_EXPORT_WITH_RC4_40_MD5	RC4_MD5_EXPORT	NO
SSL_RSA_WITH_RC4_128_MD5	RC4_MD5_US	NO
SSL_RSA_WITH_RC4_128_SHA	RC4_SHA_US	NO
SSL_RSA_WITH_DES_CBC_SHA	(v6.0 and above) DES_SHA_EXPORT (v 5.3) TLS_RSA_WITH_DES_CBC_SHA	YES
SSL_RSA_WITH_3DES_EDE_CBC_SHA	(v6.0 and above) TRIPLE_DES_SHA_US (v5.3) TLS_RSA_WITH_3DES_EDE_CBC_SHA	YES

Configure Your Server for Server Authentication

Server authentication is configured in the IBM Key Manager Tool.

1. Open the IBM Key Manager Tool. To create the key Database for the queue manager, go to **Key Database > New > key.kdb**
2. Select CMS for the Key database type.
3. Create and save an SSL key store file for the queue manager. Example: For a queue manager named MyQueueManager, the SSL key store would be located at
C:/ProgramFiles/IBMwebsphereMQ/qmgrs/MyQueueManager/ssl/key.kdb
4. Enter the password for this key database file and stash the password to a file.
5. Select **Personal Certificate Requests** from the pulldown. Click on **New**.

6. Enter the information required to get the csr request file. This file will be sent to the CA to get the signed certificate for this queue manager. Alternately, you can create a self-signed certificate instead of using CA signed certificate.

Note: While generating the csr, make sure that the title for the certificate is `ibmwebspheremqqueuemanagername` in lowercase letters. Example: `ibmwebspheremqmyqueuemanager`.

7. Before Checking in the certificate received from the CA, make sure the root certificate of the CA should be checked in as the Signer Certificate.
8. Select **Personal Certificates** from the pull-down. Click on **Receive** to add the certificate received from the CA for this queue manager. Example: `cert.der`
9. Open the IBM MQ Explorer. Right-click on the *Queue manager in use* > **Properties** > **SSL**.
10. Set the key repository to the path of the key.kdb file generated above. Example: `C:\ProgramFiles\IBM\WebSphereMQ\Qmgrs\MyQueueManager\ssl\key`

Note: No file extension is required for a kdb file.

11. Select **Queue Manager** > **Advanced** > **Channels**. Navigate to the server Connection Channel that is being used on the application side in the configuration of our adapter, and right-click it. Example: `SYSTEM_ADMIN_SVRCONN`
12. Right-click this channel, go to **SSL**, and select the correct SSL CipherSpec from the dropdown.
13. For Server Authentication, set **Authentication of parties initiating connections** to **Optional**. This completes the SSL settings for server authentication on the server side.
14. Use the IBM Key Management Tool to extract the certificate we created above from the **Personal Certificates** > **extract**. Example: The certificate is extracted as `myqueuemanager.der`
15. To import this certificate into our application (to use the certificate for the authentication of this queue manager, go to **Trading Partner** > **Digital Certificates** > **CA** in your application.
16. Add the certificate extracted above here.
17. Configure the WebSphereMQ Adapter, WsmqSuite Aysnc Adapter to set this certificate as the CA certificate. Using `wsmqSuite` to send the messages, this can be chosen from the pull-down menu using `GPM`.

Client Authentication

In client authentication, not only does the client authenticate the server, but the server also authenticates the client. To do this, repeat the above steps to create a platform for server authentication.

For client authentication, follow these steps:

1. On your application, go to **Digital Certificates** > **System Certificates**. Create a self-signed certificate, or run the certificate wizard to create a CA signed certificate.
2. Send the csr to the CA and get the client certificate.

Note: In this case, there is no restriction on the label name as it there was for server authentication.

3. The generated certificate can be viewed in the text format under System Certificates. Extract this file and import inside the key repository `key.kdb` you created above for the `queuemanager`. Import it by choosing the Signer Certificates in the IBM Key Management Tool.
4. Under **MQ Explorer** > *MyQueueManager* > **Advanced** > **Channels** > **Properties** > **SSL**, set **Authentication of parties initiating connections** to **Required**.
5. Shutdown and restart it to initiate some security updates.
6. On the SSL page in the services configuration of WebSphereMQ adapters in your application, choose the trusted certificates and Key Certificates for the client authentication.

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.
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/putAppType	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.

Field	Description
MQ/userId	The user identifier of this message.

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	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.

WebSphereMQ Suite Async Receiver Adapter

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	Related services: <ul style="list-style-type: none"> • 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	<p>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.</p> <hr/> <p>Note: The WSMQ Suite Commit service, which is the “other end” of this service, is ignored if it exists within a distributed transaction.</p> <hr/>
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 the 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 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.

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 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. <hr/> <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> <hr/> <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> <hr/> <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>
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> <hr/> <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> <hr/> <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> <hr/> <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>

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" data-bbox="850 327 1472 1262"> <thead> <tr> <th data-bbox="855 333 1159 382">Codeset name</th> <th data-bbox="1164 333 1468 382">Use CCSID</th> </tr> </thead> <tbody> <tr><td data-bbox="855 388 1159 436">ISO 8859-1</td><td data-bbox="1164 388 1468 436">819 (default)</td></tr> <tr><td data-bbox="855 443 1159 491">ISO 8859-2</td><td data-bbox="1164 443 1468 491">912</td></tr> <tr><td data-bbox="855 497 1159 546">ISO 8859-3</td><td data-bbox="1164 497 1468 546">913</td></tr> <tr><td data-bbox="855 552 1159 600">ISO 8859-5</td><td data-bbox="1164 552 1468 600">915</td></tr> <tr><td data-bbox="855 606 1159 655">ISO 8859-6</td><td data-bbox="1164 606 1468 655">1089</td></tr> <tr><td data-bbox="855 661 1159 709">ISO 8859-7</td><td data-bbox="1164 661 1468 709">813</td></tr> <tr><td data-bbox="855 716 1159 764">ISO 8859-8</td><td data-bbox="1164 716 1468 764">916</td></tr> <tr><td data-bbox="855 770 1159 819">ISO 8859-9</td><td data-bbox="1164 770 1468 819">920</td></tr> <tr><td data-bbox="855 825 1159 873">ISO 8859-13</td><td data-bbox="1164 825 1468 873">921</td></tr> <tr><td data-bbox="855 879 1159 928">ISO 8859-15</td><td data-bbox="1164 879 1468 928">923</td></tr> <tr><td data-bbox="855 934 1159 982">big5</td><td data-bbox="1164 934 1468 982">950</td></tr> <tr><td data-bbox="855 989 1159 1037">eucJP</td><td data-bbox="1164 989 1468 1037">954, 5050, 33722</td></tr> <tr><td data-bbox="855 1043 1159 1092">eucKR</td><td data-bbox="1164 1043 1468 1092">970</td></tr> <tr><td data-bbox="855 1098 1159 1146">eucTW</td><td data-bbox="1164 1098 1468 1146">964</td></tr> <tr><td data-bbox="855 1152 1159 1201">eucCN</td><td data-bbox="1164 1152 1468 1201">1383</td></tr> <tr><td data-bbox="855 1207 1159 1255">PCK</td><td data-bbox="1164 1207 1468 1255">943</td></tr> <tr><td data-bbox="855 1262 1159 1310">GBK</td><td data-bbox="1164 1262 1468 1310">1386</td></tr> <tr><td data-bbox="855 1316 1159 1365">koi8-r</td><td data-bbox="1164 1316 1468 1365">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																																						
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</p>																																						

Field	Description
	<p>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> <hr/> <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> <hr/> <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>
Automatically Commit? (wsmq_autocommit)	<p>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.</p>
Enable debug messages? (wsmq_debug)	<p>Used to turn on debugging messages for this adapter instance. Valid values are Yes and No. Default is No. Optional.</p>
SSL (SSL_SETTING_ssl_option)	<p>Whether SSL is active. Valid values:</p> <ul style="list-style-type: none"> • Must (SSL_MUST) • None (SSL_NONE)

Field	Description
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:- 66demg2sdsb: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:- 5837Measle1:1decdec: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 <hr/> Note: This parameter is required when configuring this adapter for use with CHIPS, and the required Get Type is GETONE.
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.

Field	Description
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) <hr/> 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. <hr/> 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. <hr/> Note: Only displayed if the wsmq_msgHandling parm is set to GROUP.
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.

Field	Description
Buffersize override (wsmq_buffersize)	If specified, overrides the default buffersize used when streaming data. Valid values are 0-999999999. Optional.
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)	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> <hr/> <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</p>

Field	Description
	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.
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.
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.

Parameter	Description
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.
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>
    </correlationId>
    <groupId>0000000000000000000000000000000000000000000000000000000000000000</groupId>
    <accountingToken>16010515000000B5E512BBA14EC030000000000000000B</accountingToken>
    </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 Sterling Integrator platforms
Related services	Related services: <ul style="list-style-type: none"> • 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	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, you should 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 Sterling Integrator 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 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>
      </sequence>
    </operation>
  </sequence>
  <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 Sterling Integrator platforms
Related services	<p>Related services:</p> <ul style="list-style-type: none"> • 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	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	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 Sterling Integrator 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 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)" />
        </output>
      </operation>
    </sequence>
  </sequence>
</process>
```



```

    <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 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 Sterling Integrator platforms
Related services	Related services: <ul style="list-style-type: none">• 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	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 Sterling Integrator 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>
      </output message>
    </operation>
  </sequence>
</process>
```

```

    <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="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	<p>Related services:</p> <ul style="list-style-type: none"> • 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	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. 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.
Returned status values	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 the 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>
        <!-- <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>
        <!-- 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>
  </sequence>
</process>
```



```

    </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 Sterling Integrator platforms
Related services	Related services: <ul style="list-style-type: none"> • 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	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 Sterling Integrator 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)

Field	Description
	<ul style="list-style-type: none"> • 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_corId 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_MQMO_grpId 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_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 (wmq_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 (wmq_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_grpId)	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.
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.

Field	Description
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>
```


Parameter	Description
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 Sterling Integrator platforms
Related services	<p>Related services:</p> <ul style="list-style-type: none"> • 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	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 Sterling Integrator 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)

Field	Description
	<ul style="list-style-type: none"> • 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>
</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>
    <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 Sterling Integrator platforms
Related services	Related services: <ul style="list-style-type: none">• 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	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 Sterling Integrator 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.

Field	Description																																						
Password (wsmq_password)	A user password if required to access the WebSphereMQ server. Valid value is alphanumeric. No default. Optional.																																						
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.																																						
poolName	The name of the connection pool associated with a session. Valid value is alphanumeric. Optional.																																						
poolLimit	The maximum number of connections on a queue WebSphere MQ Suite Open Session service can open when connecting to the WebSphereMQ server. After the number of connections reaches the maximum limit specified, the sessions wait until a connection is released. Valid value is any number. Optional.																																						

Field	Description
	Note: If you specify a value for poolName and do not specify a value for poolLimit, the value of poolLimit will be set to 0.
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>

  <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>

```

Business Process Example - poolName and poolLimit

The following example illustrates how the poolName and poolLimit parameters in the WebSphere MQ Suite Open Session service can be used in a business process:

```

<operation name="WebSphereMQ Suite Open Session">
  <participant name="WSMQ_OpenSession" />
  <output message="WSMQOpenSessionInputMessage">
    <assign to="wsmq_channel">channel</assign>
    <assign to="wsmq_hostname">xx.xx.xx.xxx</assign>
    <assign to="wsmq_port">1419</assign>
    <assign to="wsmq_qmanager">NewQManager</assign>
    <assign to="poolName">BPool</assign>
    <assign to="poolLimit">3</assign>
    <assign to="." from="*" /></assign>
  </output>
  <input message="inmsg" />
  <assign to="." from="*" /></assign>
  </input>
</operation>

```

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 Sterling Integrator platforms
Related services	<p>Related services:</p> <ul style="list-style-type: none"> • 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	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 Sterling Integrator 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.

Field	Description
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 0-9999999999. 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.
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 0-9999999999. 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_grpId)	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 1-9999999999. 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

Field	Description
	<ul style="list-style-type: none"> • 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.
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_putAppName)	Used to set the put application name field of the message descriptor (MQMD). Any value is valid. Optional.
Put Application Type (wsmq_MQMD_putAppType)	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.

Field	Description
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.
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)

Field	Description
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.
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.
wsmq_send_retryCount	Delay between two consecutive retries.
wsmq_send_retrySleepInterval	Length between retries for the wsmq_send_retryCount parameter. Sleep interval is in milliseconds.

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.

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.
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

Parameter	Description
	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.

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 -->
  <PrimaryDocument SCIObjectID="1893rje:1231cc1ed55:-7c55" />
</ProcessData>

<ProcessData> <!-- after PUT -->
  <PrimaryDocument SCIObjectID="1893rje:1231cc1ed55:-7c55">
    <sentMsgId>414D51204D515F47495321402102020204242ACF000009013
  </sentMsgId>
  </PrimaryDocument>
</ProcessData>
```

Business Process Example

The following example illustrates how the WebSphereMQ Suite Put Message 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>

  <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>

```

XAPI Service

The XAPI service enables you to use any XAPI provided with the application in a business process. The following table provides an overview of the XAPI service:

System Name	XAPI
Graphical Process Modeler (GPM) categories	All Services, System Services
Description	Enables use of any XAPI in a business process.
Business usage	Enables the business process writer to access any of the XAPIs and their functionality and exploit it in a business process.
Usage example	You want to see how many ACH translation maps are on your system; use the getMaplist API to find and return a list of all the maps that have "ACH" in the name. Subsequent steps in the business process might move the list to a document and export it to a folder on a server or e-mail it to the user.
Preconfigured?	No
Requires third party files?	Yes. Must refer to a XAPI included with the Application.
Platform availability	All supported Sterling Integrator platforms
Related services	None
Application requirements	Must have access to XAPIs
Initiates business processes?	No
Invocation	From a business process
Business process context considerations	After invoking a XAPI, the results are stored in the WFC so that a subsequent step in the business process can use the results.
Returned status values	Returned status values: <ul style="list-style-type: none">• Success• Error
Restrictions	None

Persistence level	Default
Testing considerations	Test by using in a business process.

How the XAPI Service Works

The XAPI service enables you to call any of the application XAPIs and return results which can then be used by a later step in the business process.

Implementing the XAPI Service

To implement the XAPI service, complete the following tasks:

1. Create a configuration of the XAPI service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the XAPI Service*.
2. Specify field settings in the GPM as necessary.

Configuring the XAPI Service

You must specify field settings at the business process level, using 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.
api	The name of the XAPI to be invoked. Required. This is for the name of the API you want to call. The XAPI framework will fail if the xapi is bad. If this parameter is not specified, then this business process must fail and an error message "Required xapi parameter is missing." must be written to the status report. Required.
xml	This is the xml content that needs to be passed to the API. If this parameter is not available then input document must be treated as the input XML. If input document is also not available, then this business process must fail and the error message "Missing Data" must be written to the status report. If the input XML is not a well formed Document object, write an error message "Invalid input" to the business process status report and this business process must fail. Required.
split	The xpath to split a document to several parts. You can use this parameter to specify several inputs to the API in the xml. You specify the split to get the different inputs. See the example above for how to use it. The resulting outputs of the call will be saved into Document-1,, Document-n. Optional.
template	Pass on this parameter to the xapi framework if specified. When specified, if the template XML is not a well formed Document object, write an error message "Invalid Template" to the business process status report and this business process must fail. Optional. Note: Some APIs ignore the template.

Field	Description
Userld	Pass on this parameter to the xapi framework. If not provided, a default value will be given. Optional.
Progld	Pass on this parameter to the xapi framework. If it is not provided, a default value will be given. Optional.

Business Process Example

The following example illustrates how the XAPI service can be used in a business process:

```
<process name="XAPITest">
  <sequence>
    <operation>
      <participant name="XAPIService"/>
      <output message="XAPIServiceInputMessage">
        <assign to="api">getServiceInstanceDetails</assign>
        <assign to="xml"><![CDATA[<root>
          <test>
            <ServiceInstance OrgKey=" " ServiceName="AssignService"/>
          </test>
          <test>
            <ServiceInstance OrgKey=" " ServiceName="AssociateBPsToDocs"/>
          </test></root>]>
        </assign>
        <assign to="split">/root/test/ServiceInstance</assign>
        <assign to="UserId">admin</assign>
        <assign to="ProgId">foo</assign>
      </output>
      <input message="XAPIServiceOutputMessage">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Parameters Passed From Service to Business Process

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

Parameter	Description
api	The name of the XAPI to be invoked. Required. This is for the name of the API you want to call. The XAPI framework will fail if the xapi is bad. If this parameter is not specified, then this business process must fail and an error message "Required xapi parameter is missing." must be written to the status report. Required.
xml	This is the xml content that needs to be passed to the API. If this parameter is not available then input document must be treated as the input XML. If input document is also not available, then this business process must fail and the error message "Missing Data" must be written to the status report. If the input XML is not a well formed Document object, write

Parameter	Description
	an error message "Invalid input" to the business process status report and this business process must fail. Required.
split	The xpath to split a document to several parts. You can use this parameter to specify several inputs to the API in the xml. You specify the split to get the different inputs. See the example above for how to use it. The resulting outputs of the call will be saved into Document-1,, Document-n. Optional.
template	Pass on this parameter to the xapi framework if specified. When specified, if the template XML is not a well formed Document object, write an error message "Invalid Template" to the business process status report and this business process must fail. Optional. Note: Some APIs ignore the template.
Userld	Pass on this parameter to the xapi framework. If not provided, a default value will be given. Optional.
Progld	Pass on this parameter to the xapi framework. If it is not provided, a default value will be given. Optional.

Parameters Passed From Business Process to Service

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

Parameter	Description
api	The name of the XAPI to be invoked. Required. This is for the name of the API you want to call. The XAPI framework will fail if the xapi is bad. If this parameter is not specified, then this business process must fail and an error message "Required xapi parameter is missing." must be written to the status report. Required.
xml	This is the xml content that needs to be passed to the API. If this parameter is not available then input document must be treated as the input XML. If input document is also not available, then this business process must fail and the error message "Missing Data" must be written to the status report. If the input XML is not a well formed Document object, write an error message "Invalid input" to the business process status report and this business process must fail. Required.
split	The xpath to split a document to several parts. You can use this parameter to specify several inputs to the API in the xml. You specify the split to get the different inputs. See the example above for how to use it. The resulting outputs of the call will be saved into Document-1,, Document-n. Optional.
template	Pass on this parameter to the xapi framework if specified. When specified, if the template XML is not a well formed

Parameter	Description
	<p>Document object, write an error message "Invalid Template" to the business process status report and this business process must fail. Optional.</p> <hr/> <p>Note: Some APIs ignore the template.</p>
UserId	<p>Pass on this parameter to the xapi framework. If not provided, a default value will be given. Optional.</p>
Progid	<p>Pass on this parameter to the xapi framework. If it is not provided, a default value will be given. Optional.</p>

XML Digital Signature Service

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 Sterling Integrator.
Platform availability	All supported Sterling Integrator 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	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 Sterling Integrator 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 Sterling Integrator, 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*.
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

1. Login to Sterling Integrator.
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	<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. <hr/> <p>Note: For more information about groups, see <i>Managing Services and Adapters</i>.</p>

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
Transforms	Optional. The required Transforms to be used when signing. If omitted when signing with enveloped signature type, the enveloped-signature Transform will be used. If omitted when signing non-XML document with enveloping signature type, the base64 Transform will be used. An error will be thrown

Parameter	Description
	<p>when the transform algorithm is invalid in xss4j.A valid example using Transform:</p> <pre data-bbox="846 331 1252 495"> <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.This parameter is required when primary document is empty.Each document node contains:</p> <ul data-bbox="846 667 1477 894" 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="846 961 1317 1682"> <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: <documents> <document> <documentID> xxx </documentID> </document></documents>

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>
```



```
</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>

    <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:

```

<processname="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>
  </sequence>
</processname>

```

```

</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>
</documents>

```

```

</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
nChQSDyNlyYYn6nRvvKl9lX8LWVNpJj
581E7/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>
```



```

    <DigestValue>
1k1M9eAUMUWM4FP40A1/YC9qS/Y=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>
UmZlQ5g4ummbZu4IRg+BHDpZeeX4ix1GXODhyG+f5tvw+4ZDKIGn8qm5wdAj3+/wJS84Lx/V
hRgjyaucMjM3qEoMAEXEWYRT
ag0PdeOmfRuffictYcTaGpPUYjDiWfKyFXAlCL5XmSeBW5sMGgy5Bi/Y0HB1gwlmU+Jsh6FsGM=
</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>

      <X509Certificate>
MIICBjCCAW8CBEB4tB0wDQYJKoZIhvcNAQEFBQAwSTElMCMGCSqGSIb3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvbTELMak
GA1UEBhMCU0cxZzARBgNVBAMTCnN1cmVuYV9yc2EwIBcNMDcwOTI1
MDcwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEkxJTAjBggqhkhk
iG9w0BCQEWFnN1cmVuYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTAlNHMRMwEQYDVOQDEwppzZXJlbnRmFfcNhmMIGfMA0GCSqGSI
b3DQEB AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+Aayih5p6WZeXMK6DmBFXS3Rd5Klp6hf
WNe7qBpwlWX+k6U2z
5q/6K+IMqZL3gsM8m2MByXsL111tu3f037/ZL1YEC0kZeTkW4i7RppvnQNQ
UrnmZRumxnPChVCuKxb4oCudaVutQ/nFZeyfAwIDA
QABMA0GCSqGSIb3DQEBBQUAA4GBAC+7glCs
TKBSURkwmB4k/SYVOOhhz3VkBX0he3r1/Vd6Qk8IlRjWQj5AT8e40gz+Vq00GvjaY
Ax70bvIGqn
yYE/VVJJOJ5Zw6Tott69Dx4AOCrmbZb96zOAJlcEI30l7U1h+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>
      </Reference>
    </SignedInfo>
    <SignatureValue>
gJ5H1D5gYydhG5NgFitWeiDs+K49CoFyauodfGG8mlvnBfCfPieul4dt4CG2/w7OxbsS5Kjy
J8+iHePiaVxdu3xYJu0ox4UkCK/wwwxvnxZwv+AlS+Kj2fwkvz6++auDOSC2Ucwe1ncwZFX5
xCSNq9wQYH12aULKsVB/bYtcvwo= </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>

          <X509Certificate>
MIICBjCCAW8CBEB4tB0wDQYJKoZIhvcNAQEFBQAwwSTElMCMGCSqGSIb3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvb
TELMakGAlUEBhMCU0cxZARBgNVBAMTCnN1cmVudV9yY2EwIBcNMDcwOTI1
MDcwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEk
xJTAjBqkqhkiG9w0BCQEFnN1cmVudV9yY2EwIBcNMDcwOTI1
Y29tbS5jb20xZAJBgNVBAYTAlNHMHRMwEQYDVQQDEwZlbnBvcn
NmIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5Klp
6hf WNe7qBpwlWX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL11ltu3f037/ZL1YEC0kZeTkW4i7RppvnQNQ
UrnmZRumxnPChVC
uKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAC+7glCs

```

```

TKBSURkwmbA4k/SYVOOhhz3VkBX0he3r1/V
d6Qk8I1RJwQj5AT8e40gz+Vq00GvjaYAx70bvIGqn
yYE/VVJJOG5Zw6Tott69Dx4AOCrmbZb96zOajlcEI30l7U1h+9+Uo2h
5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
  </X509Data>
</KeyInfo>
<Object xmlns="" Id="test">
  <test>
    <value1 ID="1">
testval1</value1>
    <value2 ID="2">
testval2</value2>
  </test>
</Object>
</Signature>

```

Example of Enveloping Signature with particular nodes

The following example shows the enveloping signature type with particular nodes:

```

<?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="#value1">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>
mb5wQZvk0lC4+YzJUQ0Q2eLlnNg=</DigestValue>
      </Reference>
    <Reference URI="#value2">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>
cITL2XfG2q9roG/XpFaoa/JSiGU=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>
Z6hxQRWv1+RZMU5UYIN06LmhFfGVtNiniIfeAmNy5TGX6SZAY5vgYbuhhZtq+LWG5nwzSQLX
Kv7TIb8N++LYgJeUtcwumCv
I6t6rl6PJusQiJZnucdpGxChrElrra2WrRZYXxaLkoSUrBJc1 pCnXscU6F0eHmKakIV0ZypdFZT4=
</SignatureValue>
    <KeyInfo>
      <KeyValue>
        <RSAKeyValue>
          <Modulus>
kFhwg4m9hjFmrlxVR3w0XmYx7fgGsoh+aelmXlzCug5gRV0t0XeSpaeoX1jXu6gacJ
Vl/p0lNs+av+iviDKmS94LDPJtjAcl7C9dzbbt39N+/2S9WBAtJGXk5M0Iu0aab50D
UfK55mUbpsZzwoVQrisW+KArn
WlbrUP5xWXsnwM=</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>

  <X509Certificate>
MIICBjCCAW8CBEB4tB0wDQYJKoZIhvcNAQEFBQAwSTElMCMGCSqGSIb3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNv
bTELMakGA1UEBhMCU0cxEzARBgNVBAMTCnN1cmVuYV9yc2EwIBcNMDcwOTI1
MDcwOTE3WhgPMjA2MjA2MjgwNzA5MTdaM
EkxJTAjbGkqhkiG9w0BCQEFnN1cmVuYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTAlNHMRMwEQYDVQQDEwpxZXJlbnF
fcNhmMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5
Klp6hf WNe7qBpw1WX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL111tu3f037/ZL1YEC0kZeTkW4i7RppvnQNO
UrnZrRumxnPC
hVCuKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAC+7glCs
TKBSURkwmBA4k/SYVOOhhz3VkBX0he3
r1/Vd6Qk8I1RjWQj5AT8e40gz+vq00GvjaYAx70bvIGqn
yYE/VVJJOJ5Zw6Tott69Dx4AOCrmbZb96zOAJlcEI3017U1h+9
+Uo2h5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
</X509Data>
</KeyInfo>
<Object xmlns="" Id="value1">
  <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
GZyaTXVlx3UTaX1C+215t0
mW4CYn4nyZpwJTbM18pRZq8tijquydg4roZz/yawz856uow3KH z+khzOuw78GzwQXVyyQymyVrQk=
</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>
2jmq715rSw0yVb/vlWAYkK/YBwk=</DigestValue>
          </Reference>
        </SignedInfo>
      <SignatureValue>
CBML9dFb/hEQXXR7oYfTuu4qit/VhUjwIfvPhSUQTQg0j+BFiTzFwNZaCJkZGswxDnSKhH1p
CuLn/Fpzl2CJpNduDU0Ff0pstd7MITS010/IvhDVS+Tf6WiYkN8UYTckJeg063zlbW+15mR1
Z25jCs0gW09qEsthX34qXRi7ii0= </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/V2fqo=</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>
      </Reference>
    </SignedInfo>
  </Signature>
</test>
```

```
</Transforms>
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
  <DigestValue>
nn7t7PJs5RqDp1BKZ4j1BxhX2ik=</DigestValue>
  </Reference>
</SignedInfo>
  <SignatureValue>
Ui7XYcZNkXG+9OqNgKrcRJoyCuhpoRtVmFnXVOMf8aAuGXZw3FwFxz7VLKv9c1K8ZUNW9vCs
G4EpahlCS4A
cpbVBwv00HvkhA11/tqYYB9kRK/wM4cb6sN5ULbQ4Ab0j9xyFKOQ6sr2MawOx
fdNEes6XAHbpWZvxKDR4vWxAFnE= </SignatureValue>
  <Object xmlns="" Encoding="base64" Id="sgconrad:-56000361:115d676b12e:-7988">
dGhpcyBpcyBhbiBpbnZhbGlkIHRLc3QgZG9jIGZvcjB4bWwgZHNpZyBzZXJ2aWNlLgo=</Object>
  </Signature>
</test>
```


XML Encoder Service

Note: If the input document character encoding is specified in Sterling Integrator, 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 Sterling Integrator platforms
Related services	No
Application requirements	The map specified in the map_name parameter must have been registered with Sterling Integrator 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	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. Sterling Integrator 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*.
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 Sterling Integrator. 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:

Field	Description
	<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.
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	Usage examples: <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 Sterling Integrator 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'"/>
    <assign to="." from="*" />
  </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'"/>
    <assign to="." from="*" />
  </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[@tp:certId=
        //DecryptCertId/text()/ds:KeyInfo/ds:KeyName)"/>
    <assign to="encMode" from="'DECRYPT'"/>
    <assign to="encType" from="'DOCUMENT'"/>
    <assign to="." from="*" />
  </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'"/>
    <assign to="." from="*" />
  </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 Sterling Integrator 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 Sterling Integrator.</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	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. See the Workflow Context Status report for further details.
Restrictions	No
Persistence level	None
Testing considerations	None

Requirements

The DTDs and schemas are checked in Sterling Integrator. 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*.
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 Sterling Integrator and using BPML. See *Creating a Service Configuration*.

Sterling Integrator Configuration

The following table describes the fields used to configure the XML Validation service in Sterling Integrator:

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. <hr/> 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 <hr/> 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

Field	Description
	<p>Note: This parameter is set in the Sterling Integrator configuration, but can be overwritten using BPML.</p>
xml_input_from	<p>Specifies the location of the XML to validate. Optional. Valid values:</p> <ul style="list-style-type: none"> • Procdata – Validate the XML in process data • Primdoc – Validate the XML in the primary document <p>Default is Primdoc.</p>
input_pd_xpath	<p>Identifies the XPath location of the XML to validate. Required if xml_input_from is set to Procdata.</p>

XSLT Service

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 Sterling Integrator. 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 Sterling Integrator 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	Returned status values: <ul style="list-style-type: none"> • Basic status Success – Transformation was successful.

	<ul style="list-style-type: none"> • 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. Sterling Integrator 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

Sterling Integrator 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_params	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_params 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

Field	Description
	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.

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 Sterling Integrator, you must first check them in.

To check in an XSLT style sheet to Sterling Integrator:

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.

Note: 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 Sterling Integrator 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!**

Sterling Integrator 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!**

Sterling Integrator returns a list of matches unless no XSLT style sheets meet your criteria.

About Search Results

When you search for an XSLT style sheet, Sterling Integrator 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 Sterling Integrator, you can edit it without checking it out of Sterling Integrator.

To edit an XSLT style sheet in Sterling Integrator:

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 Sterling Integrator, 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 Sterling Integrator:

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.

Note: 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. Sterling Integrator 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 Sterling Integrator and prevent anyone from modifying the file while you are making changes, you check out a version from Sterling Integrator. 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 Sterling Integrator:

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 Sterling Integrator 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**. Sterling Integrator 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 Sterling Integrator services and business processes.

You can enable or disable an XSLT style sheet in two ways:

- At the time you check it in to Sterling Integrator
- 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

The Yantra adapters enable you to use Yantra functionality from within Sterling Integrator. The following table provides an overview of the Yantra adapters:

System Name	These adapters are dynamically created when Sterling Integrator 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. Sterling Integrator 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 Sterling Integrator.
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 Sterling Integrator 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

	theSterling Integrator database, you may also need to load the appropriate database jar using the install3rdParty script (for example, instance classes12.zip for Oracle).
Platform availability	Available for: <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	Returned status values: <ul style="list-style-type: none"> • 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 Sterling Integrator is started. Yantra functionality is made available in Sterling Integrator through Yantra API groups.

When Sterling Integrator 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 Sterling Integrator 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

To install 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 Sterling Integrator 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 Sterling Integrator.
3. From the bin directory where Sterling Integrator is installed, install the .jar files by running the install3rdParty script included with Sterling Integrator. 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 Sterling Integrator 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 Sterling Integrator.

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 Sterling Integrator 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 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 Yantra 5.x or 7.x version of Sterling Multi-Channel Selling and Fulfillment (MCF) 8.0 or 8.2:

```
<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>
```

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 you have integrated with Sterling MCF 8.5:

```
<process name = "YantraTest">
  <operation name="Get Inventory Supply">
    <participant name="<instancename>" />
    <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>
```

Messages Generated By Yantra Startup Class

The following messages are in noapp.log if the yfsclient.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 Sterling Integrator starts, it will check for Yantra.

Suppressing the Messages

If you stop using Yantra with Sterling Integrator, 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 Sterling Integrator 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.
```

Zengin TCP/IP Adapter

Zengin is a standard Japanese communication protocol. The Zengin TCP/IP adapter enables Sterling Integrator 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	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. 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. 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.
Preconfigured?	No
Requires third party files?	No
Platform availability	Available for: <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	The Zengin TCP/IP adapter supports Zengin TCP/IP version 2. The application must provide at least one document in process data if a “send” action is specified.
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*.

When the Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator 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*.
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*.

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*.
2. Create a Zengin Configuration Import service configuration. For information, see *Implementing the Zengin Configuration Import Service Configuration*.
3. Configure the Zengin XML configuration file. For information, see *Configuring the Zengin XML Configuration File*.
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*.
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* 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 *Creating a Service Configuration*.
2. Configure the Zengin Configuration Import service. For information, see *Configuring the Zengin Configuration Import Service*.
3. Create a File System adapter configuration to read the Zengin XML configuration file. For information, see *Configuring the File System Adapter*.
4. Create the Zengin TCP/IP adapter configuration. For information, see *Creating a Service Configuration*.
5. Configure the Zengin TCP/IP adapter. For information, see *Configuring the Zengin TCP/IP Adapter*.
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 Sterling Integrator.

Sterling Integrator Configuration

The following table describes the fields used to configure the Zengin Configuration Import service in Sterling Integrator:

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	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.

Field	Description
	<ul style="list-style-type: none"> • 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. <hr/> <p>Note: For more information about groups, see <i>Using Service Groups</i>.</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 Sterling Integrator and in the GPM.

Sterling Integrator Configuration

The following table describes the fields used to configure the required File System adapter instance in Sterling Integrator:

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.
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. <hr/> <p>Note: For more information about groups, see <i>Using Service Groups</i>.</p>
Collection Folder (collectionFolder)	Any folder or subfolder (on the same computer where Sterling Integrator is installed) where files are collected. 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.
Filename Filter (filter)	Leave blank

Field	Description
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>
</sequence>
</process>
```

```

</operation>
<operation name="Zengin Configuration Import Service">
  <particpant 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.bpm1) 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 Sterling Integrator. Find this file in the Sterling Integrator 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 Sterling Integrator 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 localhost. 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 Sterling Integrator 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.

XML Element	Description
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	<p>Length of the record in bytes. When sending a file or responding to a request for a file:</p> <ul style="list-style-type: none"> record_length is required. If fixed_record_length is set to true, then $1 < record_length \leq 2043$, and record_length must be a factor of file size. If fixed_record_length is set to false, then $0 < record_length \leq 2041$. <p>When requesting a file:</p> <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. <p>When receiving a file, the record_length value is not used.</p>
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.
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(Sterling Integrator)-->
    <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>FFFFFFFFFFFFFF</password>
  <file_definition>
    <file_alias>send_variable_uncompr</file_alias>
    <filename>FOFOFOFOFOFOFOFOFOFOFOFO</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>000000000000000000000001</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.

Sterling Integrator 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 Sterling Integrator 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 Sterling Integrator. 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 Sterling Integrator 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 Sterling Integrator and in the Graphical Process Modeler. For general information about service and adapter configurations, see *Creating a Service Configuration*.

Configuring the Zengin Sender and Requester Adapter

Sterling Integrator 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 Sterling Integrator:

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. <p>Note: For more information about groups, see <i>Using Service Groups</i>.</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: <myDocs> <FileAlias1 SCIOBJECTID="someDbObjectId"/> <FileAlias2 SCIOBJECTID="anotherDbObjectId"/> </myDocs> and the business process contains the following:<assign to="ZenginDocuments">//myDocs/*</assign> 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: <assign to="ZenginFileAlias">myAlias</assign> In this case, you must configure myAlias as a file alias using the Zengin XML configuration file.</p>
ZenginHostname	Remote Zengin server IP address or DNS name. Must correspond with the server specified by ZenginRemoteServerAlias. If left blank in the Sterling Integrator 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 Sterling Integrator 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 <i>//client/zengin_id/alias</i> element in the Zengin XML configuration file. If left blank in the Sterling Integrator configuration, you must specify this parameter either here or at runtime in process data.
ZenginVariableRecLenPreInserted	<p>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.</p> <hr/> <p>Note: This field is no longer necessary, but is retained for compatibility with older files preceded with a 2-byte record length.</p> <hr/>

Configuring the Zengin Receiver and Responder Adapter

Sterling Integrator 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 Sterling Integrator:

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 Sterling Integrator.
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 AutoRetry 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 AutoRetry is set to true and MaxRetry 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.
<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.

Property	Description
<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/bin/setupfiles.sh` (UNIX)
 - `install_dir\bin\setupfiles.cmd` (Windows)
6. Stop and restart Sterling Integrator 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>
      </output>
    </operation>
  </sequence>
</process>
```

```

    <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>
        <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>
    </operation>
  </sequence>
</process>

```

```

</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>
<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>

```

```

</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>

<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 Sterling Integrator 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>
```

Copyright

Licensed Materials - Property of Sterling Commerce

© Copyright Sterling Commerce, an IBM Company 2000, 2011 All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by contract with Sterling Commerce

Additional copyright information is located on the Sterling Integrator 5.1 Documentation Library:

<http://www.sterlingcommerce.com/Documentation/SI51/CopyrightPage.htm>

Index

A

adapter

- Adapter for PeopleSoft 180
 - Adapter for PeopleSoft CRM CIC 222
 - MSMQ 92
 - OFTP 120, 134, 145
 - Oracle AQ JMS Queue 152
 - Oracle AQ JMS Topic 156
 - SAP Suite 279
 - SAP XI 257
 - Script 354
 - SMTP Send 421
 - SNMP Trap 437
 - Sterling Information Broker 413
 - third-party 356
 - TIBCO Rendezvous 519
 - Vitria (BusinessWare) 538
 - webMethods 548
 - WebSphere MQ 566
 - WebSphereMQ Suite Async Receiver 590
 - Yantra 691
 - Zengin 698
- Adapter for PeopleSoft
- configuring Translation service for PS Send 188
 - Consume service 192
 - creating a typing map for PS Send 187
 - creating translation map for PS Send 188
 - description 180
 - DXREF service 195
 - how it works 182
 - HTTP Client adapter 191
 - implementing 185
 - implementing PS Send 185
 - Inflation service 193
 - Inflation service configuration for PS Receive 193
 - PeopleSoft time stamp 197
 - ping request 185
 - PS Receive 183, 195
 - PS Send 182
 - PS Send and Wait 183
 - Receive 193, 197
 - Request Response XREF 191
 - Request Response XREF service 191
 - requirements 182
 - Send and Wait 191, 192
 - Translation service 191
 - Typing service 185
 - Typing service for PS Receive 193
 - Typing service for PS Send 187
- Adapter for PeopleSoft CRM CIC
- AIX requirements 223
 - building DTDs 228
 - configuring 230

Adapter for PeopleSoft CRM CIC (*continued*)

- connection management 231
 - create operation 225
 - data structure 224
 - delete operation 227
 - description 222
 - error messages 229
 - fields 225
 - how it works 223
 - HP-UX requirements 223
 - implementing 229
 - installing 230
 - object hierarchy 224
 - object type 224
 - operations 225
 - read operation 226
 - record sets 225
 - requirements 223
 - update operation 227
- Adding document to process data
- example 514
- advanced statuses 314
- Application adapters
- Adapter for PeopleSoft 180
 - Adapter for PeopleSoft CRM CIC 222
 - SAP Suite 279
 - TIBCO Rendezvous 519
 - Vitria (BusinessWare) 538

B

- Bean Scripting Framework 354
- Bean scripts, using with Script adapter 354
- BPML example
 - MIME service 76
 - SMTP Send adapter 424
- business process ID
 - putting into process data 512
- Business scenario
 - SAP Suite adapter 282
 - SMTP Send adapter 423
- Business Scenario
 - MIME service 75
- BW30.jar file 538

C

- client50.jar file 556
- Consume service
 - using with Adapter for PeopleSoft 189
- crackRequest
 - MIME service 83
 - MIME service parameters 82

D

- Decision Engine service
 - using with Adapter for PeopleSoft 192, 194
- document ID
 - putting into process data 511
- document name
 - putting into process data 513
- DTD 187, 225, 229

E

- end time
 - putting into process data 509
- Examples
 - MIME service 75, 83, 85, 87
 - Release service 240, 242, 243
 - SAP Suite adapter 282, 284, 323
 - SAP Suite adapter (inbound) 324
 - SAP Suite adapter (outbound) 333
 - SMTP Send adapter 423, 432, 434, 436
 - This service 508, 509, 510, 511, 512, 513, 514, 516, 517
- exporting or importing a Zengin XML configuration 707
- extended rule 187
- Extracting MIME Decoded Documents
 - example 87

F

- File System adapter
 - use with MIME service 75
- Formatting times
 - example 510
- free text, SMTP Send adapter 422

G

- GPM Example
 - MIME service 79
 - SMTP Send adapter 424

H

- HandleSOAPRequest predefined business process 473
- HTTP Respond service
 - using with Adapter for PeopleSoft 192
 - using with Adapter for PeopleSoft PS Receive 197
- HTTP Server adapter
 - using with Adapter for PeopleSoft 192

I

- Inflation service
 - using with Adapter for PeopleSoft 192
- install3rdParty.sh script 230, 291, 292
 - using for SAP Suite adapter 291, 292
 - using with Adapter for PeopleSoft CRM CIC 230

M

- Mail Delete service
 - configuring 29
 - implementing 29, 32
- Mail Mime service 6, 7, 8
 - configuring 8
 - how it works 7
 - implementing 7
- Mailbox Add service
 - configuring 16
 - how it works 15
 - implementing 16
- Mailbox Correlate Document service
 - how it works 20
- Mailbox Delete Mailbox service 21, 22, 23, 24, 25
 - business process example 24
 - configuring 23
 - how it works 22
 - implementing 23
 - overview 21
 - parameters passed from service to BP 25
- Mailbox Delete service
 - how it works 28
- Mailbox Evaluate All Automatic Routing Rules
 - configuring 33
- Mailbox Evaluate All Automatic Routing Rules service
 - how it works 32
- Mailbox Evaluate Routing Rule service
 - configuring 36
 - implementing 36
- Mailbox Evaluate Routing Rules service
 - how it works 36
- Mailbox Extract Abort service
 - configuring 42
 - how it works 41
 - implementing 42
- Mailbox Extract Begin service
 - configuring 46, 50
 - how it works 45
 - implementing 46
- Mailbox Extract Commit service
 - how it works 49
 - implementing 50
- Mailbox Query service
 - configuring 56
 - how it works 55
 - implementing 56
 - overview 54
 - parameters passed from BP to service 58
 - parameters passed from service to BP 57
- Mailbox Scheduled Delete service
 - configuring 62
 - how it works 62
 - implementing 62
- Mailbox Update service
 - configuring 66
 - how it works 66
 - implementing 66

- Mapped Extraction service
 - business process example 70
 - how it works 69
 - implementing 69
 - overview 68
- MergeDocument service
 - business process example 72
 - how it works 72
 - overview 71
 - parameters passed from service to BP 72, 73
- MIME Decoding
 - example 85
- MIME Encoding
 - example 83
- MIME multipart document
 - body 74, 83
 - header 74, 83
 - message 74, 83
- MIME service
 - BPML example 76
 - business scenario 75
 - configuring 80
 - crackRequest 83
 - crackRequest parameters 82
 - description 74
 - extracting decoded documents 87
 - GPM example 79
 - implementing 80
 - MIME decoding example 85
 - MIME encoding example 83
 - packRequest 83
 - packRequest parameters 82
 - RFC 822-compliant document 74
 - usage example 75, 83, 85, 87
 - use with File System adapter 75
- MSMQ adapter 92, 93, 94
 - description 92
 - how it works 93
 - implementing 94
 - requirements 93
- MSMQ Send service
 - business process example 103
 - GPM configuration 102
 - implementing 100
 - output from service to business process 103
 - overview 99
- MSMQPrime
 - installing 100
- MSMQPrime as Windows service
 - installing 101
 - uninstalling 102

N

- Network Report Reconciliation service
 - configuring 107
 - description 105
 - how it works 106
 - implementing 106

- Non-raw message mode
 - example 432

O

- Obscure Data - Obscure Primary Document service
 - description 112
- Obscure Data - Obscure Process Data Values service
 - configuring 116
 - implementing 116
- Obscure Data - Reveal Primary Document service
 - description 109
- OFTP adapter
 - before using 132, 151
 - configuring 123, 136, 144, 149
 - description 120, 134, 145
 - implementing 123, 144, 148
 - license activation 123, 148
 - Mailbox mode requirements 123
 - manual mode requirements 123
 - requirements 144
- Oracle AQ JMS Queue adapter
 - description 152
 - how it works 152
 - implementing 153
- Oracle AQ JMS Topic adapter
 - configuring 157
 - description 156
 - how it works 156
 - implementing 157
- Oracle E-Business Suite Adapter Configuration service 160, 161
 - configuring 161
 - implementing 161
- Oracle E-Business Suite Message service 165, 166
 - configuring 166
 - implementing 166

P

- packRequest
 - MIME service 83
 - MIME service parameters 82
- PeopleSoft
 - Adapter for PeopleSoft 180
 - Adapter for PeopleSoft CRM CIC 222
 - Timestamp Service for PeopleSoft 525
- PeopleSoft adapter
 - PS Receive 183
- PGP Package service
 - configuring 201
 - implementing 201
 - overview 198
- PGP Unpackage service
 - configuring 215
 - implementing 215
 - overview 212
- ping reply 195
- predefined business process
 - HandleSOAPRequest 473

predefined business process (*continued*)

SAP Inbound Delivery 281, 300

SAP Inbound IDoc 281, 300

SAP Outbound IDoc 281, 300

SendSOAPRequest 478, 484

Pretty Good Privacy (PGP)

using 198, 212

Produce service

using with Adapter for PeopleSoft 192

R

raw message format 434

example 434

Release service

best practices 245

how it works 240

implementing 240

overview 239

usage example 240, 242, 243

Releasing document from process data

example 517

Removing primary document reference

example 242

Removing temporary document

example 243

Removing temporary variables

example 240

Report service

configuring 247

implementing 247

Request Response DXREF service

using with Adapter for PeopleSoft 192

using with Adapter for PeopleSoft PS Receive 195

Request Response XREF service

using with Adapter for PeopleSoft 189

Restore Business Process service

configuring 249

how it works 249

implementing 249

overview 248

RFC 822-compliant document

MIME service 74

RFC822-compliant document

raw message format 434

SMTP Send adapter 422

RMI adapter

.jar files 256

architecture 253

asynchronous requests 252

business process 254

business process requests 254

configuring 254

connections 254

developing client code 256

how it works 252

implementing 253

requirements 252

synchronous requests 253

S

SAP Suite adapter 279, 280, 281, 282, 284, 289, 291, 292, 293, 294, 295, 296, 297, 300, 303, 305, 306, 310, 312, 313, 314, 323, 324, 333

ALE-based IDoc RFC 296

business process example 284

business processes 281, 300

business scenario 282

communicating and processing IDocs 281

configuration example 282

configuring 293

configuring inbound and outbound cross-references 305

connection pooling 300

creating configuration 293

data extraction 294

description 279

error messages 310

exporting parameters 313

file-based IDoc RFC 295

how it works 280

implementing 289

inbound processing example 324

installing on iSeries 292

installing on UNIX or Windows 291

integration mode 294

load-balancing 279

outbound process example 333

overview 279

retrieving IDoc format descriptions 306

RFC 294

supported platforms 280, 289

synchronous BAPI 294

synchronous RFC 295

translating IDocs to EDI 281, 303, 305

usage example 282, 323

user properties 297

workflow definition parameters 312

workflow transaction definition parameters 310

SAP XI adapter

business process example 270

configuring 263

how it works 259

implementing 262

overview 257

parameters passed from BP to service 277

Script adapter

configuring 356

description 354

how it works 355

implementing 356

Securing the Zengin XML Configuration file 706

SendSOAPRequest predefined business process 478, 484

service

Mail Mime 6

Mailbox Add 14

Mailbox Correlate Document 19

Mailbox Delete 27

Mailbox Delete Mailbox 21

Mailbox Evaluate All Automatic Routing Rules 31

service (*continued*)

- Mailbox Evaluate Routing Rule 35
- Mailbox Extract Abort 40
- Mailbox Extract Begin 44
- Mailbox Extract Commit 48
- Mailbox List 52
- Mailbox Query 54
- Mailbox Scheduled Delete 61
- Mailbox Update 65
- Mapped Extraction 68
- MergeDocument 71
- MIME 74
- Network Report Reconciliation 105
- Obscure Data - Obscure Primary Document 112
- Obscure Data - Reveal Primary Document 109
- Oracle E-Business Suite Adapter Configuration 160
- Oracle E-Business Suite Message 165
- PGP Package service 198
- PGP Unpackage service 212
- Release 239
- Restore Business Process 248
- Sleep 418
- SOA Outbound Message Processing 445, 451, 457, 463
- SOAP Inbound 470
- SOAP Outbound 482
- This 504
- Timestamp for PeopleSoft 525
- Timestamp Utility 527
- Timezone Offset 530
- Typing 532
- User 535
- Wait 546
- Wait Notify 544
- WebSphereMQ Suite Backout 602
- WebSphereMQ Suite Close Queue 606
- WebSphereMQ Suite Close Session 610
- WebSphereMQ Suite Commit 614
- WebSphereMQ Suite Get Message 618
- WebSphereMQ Suite Open Queue 627
- WebSphereMQ Suite Open Session 632
- WebSphereMQ Suite Put Message 638
- XML Encoder 673
- XML Encryption 676
- XML Validation 680
- XSLT 684
- ZlibInflate 717
- SFTP Client adapter
 - implementing 360
- SFTP Client all services
 - business process example 362
- SFTP Client Begin Session service
 - business process example 375
 - output from business process to service 370
 - output from service to business process 374
- SFTP Client CD service
 - business process example 378
 - output from business process to service 377
 - output from service to business process 377
- SFTP Client DELETE service
 - business process example 381

SFTP Client DELETE service (*continued*)

- output from business process to service 380
- output from service to business process 380

SFTP Client End Session service

- business process example 384
- output from business process to service 383
- output from service to business process 383

SFTP Client LIST service

- business process example 390
- output from business process to service 389
- output from service to business process 389

SFTP Client MOVE service

- business process example 394
- output from business process to service 393
- output from service to business process 393

SFTP Client PUT service

- business process example 397
- output from business process to service 396
- output from service to business process 396

SFTP Client PWD service

- business process example 400
- output from business process to service 400
- output from service to business process 400

SFTPClientDemoAllServices

- BPML code 362
- graphical model 362

Simple Object Access Protocol (SOAP) Outbound service 482

Sleep service

- business process example 419
- configuring 418
- overview 418
- parameters passed from BP to service 420

SMTP Send adapter

- B2B mode 436
- BPML example 424
- business scenario 423
- configuring 425
- free text 422
- GPM example 424
- non-raw message mode 432
- raw message format 434
- RFC822-compliant document 422
- usage example 423, 432, 434, 436

SNMP Trap adapter

- architecture 438
- configuring 440
- configuring as trap generator 440
- configuring as trap receiver 443
- description 437
- how it works as a trap generator 438
- how it works as a trap receiver 441
- implementing as trap generator 439
- implementing as trap receiver 443
- requirements 438
- trap generator 437
- trap receiver 437

SOA Inbound Security

- parameters provided in Request Security Settings (GIS Provider) 454

- SOA Inbound Security service
 - business process examples 454
 - configuring 452
 - implementing 452
 - overview 451
 - parameters passed from bp to service 454
 - parameters passed from service to bp 453, 454
- SOA Outbound Message Processing service
 - business process examples 447, 459
 - configuring 446, 458
 - implementing 446, 458
 - overview 445, 457
 - parameters passed from service to bp 450, 462
- SOA Outbound Security Processing service
 - business process examples 466
 - configuring 464
 - GPM parameters 465
 - implementing 464
 - overview 463
 - parameters passed from service to bp 466
- SOAP Inbound service
 - configuring 472
 - description 470
 - implementing 472
- SOAP outbound service
 - overview 482
- SSH Key Grabber Adapter 487
- SSLCertGrabber adapter
 - implementing 493
- start time
 - putting into process data 509
- Sterling Control Center service
 - configuring 497
 - implementing 497
- Sterling Information Broker adapter
 - configuring 414
 - description 413
 - how it works 414
 - implementing 414

T

- TCP/IP, Zengin adapter 698
- Text Document Appender service
 - overview 502
- third-party software
 - adapters for using 356
- This service
 - add document to process data 514
 - format time 510
 - get business process ID 512
 - get document ID 511
 - get document name 513
 - get start and end time 509
 - get time difference 516
 - how it works 506
 - overview 504
 - release document from process data 517
 - usage example 508

- TIBCO Rendezvous adapter
 - configuring 520
 - data flow 520
 - description 519
 - how it works 519
 - implementing 520
 - publisher transaction 522
 - required files 521
 - subscriber transaction 523
- tibrvj.jar file 521
- TID management outbound 313
- Timestamp Service for PeopleSoft
 - overview 525
 - using with Adapter for PeopleSoft 192
- Timestamp Utility service
 - business process examples 528
 - configuring 528
 - how it works 527
 - implementing 528
- Timezone Offset service 530, 531
 - implementing 531
- Translation service
 - using with Adapter for PeopleSoft 189, 192
- Typing service
 - configuring 533
 - description 532
 - how it works 532
 - implementing 533
 - using with Adapter for PeopleSoft 189, 192

U

- User service
 - configuring 535
 - description 535
 - how it works 535
 - implementing 535
 - output results 536

V

- VanAPI library 224
- vanjavi.jar file 230
- Vantive 222
- Vitria (BusinessWare) adapter
 - configuring 540
 - data flow 539
 - description 538
 - how it works 539
 - implementing 540
 - publisher transaction 541
 - required files 538
 - subscriber transaction 542

W

- Wait Notify service 544
- Wait service 546, 547
 - how it works 547

- Wait service (*continued*)
 - implementing 547
 - weblogic.jar file 252
 - webMethods adapter
 - basic operations 549
 - components 548
 - description 548
 - DTD generator 564
 - how it works 548
 - implementing Publisher adapter 556
 - implementing Subscriber adapter 559
 - installing 556
 - internationalization 565
 - structure of data 552
 - webMethods Publisher adapter
 - configuring 556
 - webMethods Subscribe adapter
 - configuring 560
 - WebSphere MQ adapter
 - description 567
 - how it works 567
 - implementing 568
 - installing 569
 - WebSphere MQ Adapter
 - configuring 570
 - WebSphereMQ Suite
 - examples 600
 - WebSphereMQ Suite Async Receiver adapter
 - configuring 591
 - example 600
 - implementing 591
 - overview 590
 - parameters passed from service to bp 598
 - WebSphereMQ Suite Backout service
 - business process example 604
 - configuring 603
 - implementing 603
 - overview 602
 - parameters passed from bp to service 603
 - parameters passed from service to bp 603
 - WebSphereMQ Suite Close Queue service
 - business process example 608
 - GPM settings 607
 - implementing 607
 - overview 606
 - parameters passed from bp to service 607
 - WebSphereMQ Suite Close Session service
 - business process example 611
 - configuring 611
 - implementing 611
 - overview 610
 - parameters passed from bp to service 611
 - WebSphereMQ Suite Commit service
 - business process example 616
 - configuring 615
 - implementing 615
 - overview 614
 - parameters passed from bp to service 615
 - parameters passed from service to bp 615
 - WebSphereMQ Suite Get Message service
 - business process example 625
 - GPM settings 619
 - implementing 619
 - overview 618
 - parameters passed from bp to service 624
 - parameters passed from service to bp 623
 - process data example 622
 - WebSphereMQ Suite Open Queue service
 - business process example 629
 - GPM settings 628
 - implementing 628
 - overview 627
 - parameters passed from bp to service 629
 - parameters passed from service to bp 629
 - WebSphereMQ Suite Open Session service
 - business process example 635
 - GPM settings 633
 - implementing 633
 - overview 632
 - parameters passed from bp to service 635
 - parameters passed from service to bp 635
 - WebSphereMQ Suite Put Message service
 - business process example 644
 - GPM settings 639
 - implementing 639
 - overview 638
 - parameters passed from bp to service 643
 - parameters passed from service to bp 643
 - WSDL service
 - configuring 588
 - how it works 588
 - implementing 588
- ## X
- XML configuration file
 - configuring 704
 - implementing for the Zengin adapter 701
 - Zengin adapter configuration 701
 - XML Digital Signature service
 - business process example 656
 - implementing 653
 - output from business process to service 654
 - overview 652
 - XML Encoder service
 - implementing 674
 - overview 673
 - XML Encryption service 676, 678
 - implementing 678
 - overview 676
 - XML Validation service
 - configuring 682
 - how it works 681
 - implementing 681
 - requirements 681
 - XREF cookie 190, 193, 194
 - XSLT service
 - configuring 685

XSLT service (*continued*)
description 684
implementing 685
managing XSLT stylesheets 686
requirements 685

Y

Yantra adapter
business process example 1 695
business process example 2 695
configuring 694
how it works 692
implementing 693
overview 691
parameters passed from BP to service 694
parameters passed from service to BP 694

Z

Zengin adapter
Configuration Import service 699
configuring 707
configuring the XML configuration file 704
how it works 699
implementing 700
license activation 701

Zengin adapter (*continued*)
overview 698
Receiver and Responder 698
Sender and Requester 698
TCP/IP 698
XML configuration file 701
XML configuration file, configuring 704
Zengin Configuration Import service
configuring for the Zengin adapter 701
implementing 701
Zengin Import service, configuring 701
Zengin Receiver and Responder adapter
receiving a file 700
responding to a file request 700
Zengin Receiver and Responder adapter, configuration parameters
710
Zengin Sender and Requester adapter
requesting a file 700
sending a file 699
Zengin Sender and Requester adapter, configuration parameters
708
Zengin XML Configuration file 706
Zengin XML Configuration Import service
configuring 701, 702
zenginConfigHelper utility 707
ZlibInflate service
business process example 718
overview 717