Sterling Integrator

# Services and Adapters (A to L)

*Version 5.1*

IBM

Sterling Integrator

# Services and Adapters (A to L)

*Version 5.1*

IBM

# Contents

# Chapter 1. AFT Route Progress Event Reporting Service

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

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

## How the AFT Route Progress Event Reporting Service Works

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

## Implementing the AFT Route Progress Event Reporting Service

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

## Configuring the AFT Route Progress Event Reporting Service

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

## Business Process Example

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

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

## Parameters Passed From Service to Business Process

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

## Parameters Passed From Business Process to Service

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

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

# Chapter 2. Alert Service

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

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

## How the Alert Service Works

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

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

The Alert service consists of:

- A subsystem controller—the Main Alerter

- Individual instances of alerter and filter components
- A Sterling Integrator Alert service
- A default Sterling Integrator Alert Notification business process

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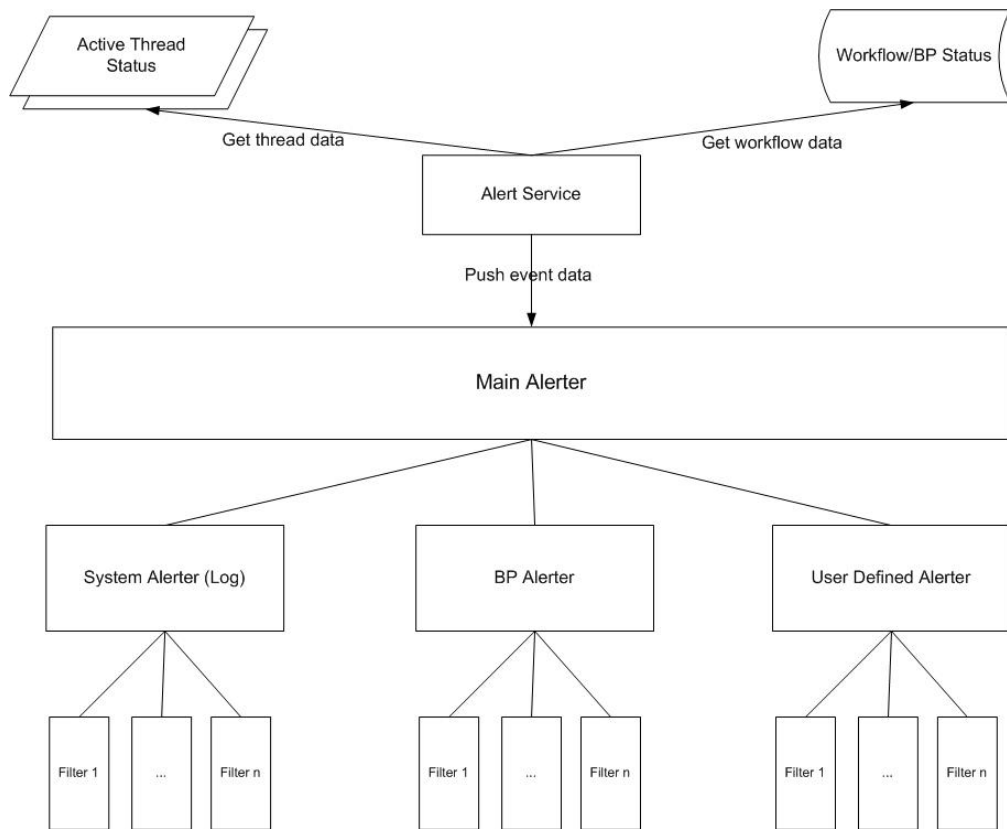


## Main Alerter

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

## Alerters

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

The system alerter is the fallback mechanism that writes errors to the Alert service log file. The business process alerter is the default mechanism for executing a business process that sends notification. You can add more alerter instances

through Sterling Integrator and more alerter classes by modifying the alerter.properties file.

## Filters

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

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

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

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

## Alert Service

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

## Alert Notification Business Processes

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

## Implementing the Alert Service

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

You can customize the Alert service using the following options:
- Modifying the predefined AlertNotification business process
- Configuring additional alerters and filter instances for the predefined classes

- Implementing new filter and alerter classes to plug in

To implement the Alert service, complete the following tasks:

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

## Modifying the Predefined AlertNotification Business Processes

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

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

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

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

```
#
# BR Alerter
#
...
alerter.2.props.1.key=process
alerter.2.props.1.value=AlertNotification
  (this is the business process to run)
alerter.2.props.2.key=load_status_rpt_and_process_data
alerter.2.props.2.value=true
  (set this value to true if you want to load the status report
into the process data)
```

1. Stop Sterling Integrator.
2. Run setupfiles.sh (UNIX) or setupfiles.cmd (Windows).
3. Restart Sterling Integrator.
4. Run setup_alerters.sh (UNIX) or setup_alerters.cmd (Windows).

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

## Configuring Additional Alerters and Filter Instances for Predefined Classes

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

The alerter.properties file has the following general form:

```
# Alerters
alerter.#.id=<alerter_name>
alerter.#.class=<fully qualified class name>
# Alerter supported keys
# - multiple_filters_logic = [AND | OR]
#
# BP Alerter
# - process = <process_name>
#
# System Alerter
# - logger = <logger_name>
# alerter.#.props.#.key=<key name>
alerter.#.props.#.value=<value for key with corresponding number>
alerter.#.filter.#.name=<filter name>
# Filters
filter.#.class=<fully qualified class name>
filter.#.name=<unique filter name>
filter.#.criteria=<filter criteria data>
```

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

The alerter.properties file has the following default configurations:

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

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

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

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

The alerter.properties file has the following default properties:

```
####################
# Alerters settings #
####################
#
# System Alerter
#
# class implementations
alerter.1.class=com.sterlingcommerce.woodstock.alert.impl.SystemAlerter
# alerter's id (must be unique)
alerter.1.id=system_alerter
# key for first value of alerter properties
alerter.1.props.1.key=logger
# Value for first property of alerter properties
alerter.1.props.1.value=alerterlogger
alerter.1.props.2.key=multiple_filters_logic
# for this property the default value is "OR", the only other you can
```

```
    specify is "AND"
alerter.1.props.2.value=AND
# filter name
alerter.1.filter.1.name=xpathfilter_type_alerter
#
# BR Alerter
#
alerter.2.class=com.sterlingcommerce.woodstock.alert.
impl.ProcessLauncherAlerter
alerter.2.id=bp_alerter
alerter.2.props.1.key=process
alerter.2.props.1.value=AlertNotification
alerter.2.props.2.key=multiple_filters_logic
alerter.2.props.2.value=OR
alerter.2.filter.1.name=xpathfilter_bp_status_notSuccess
####################
#  Filters settings #
####################
#
# Filter for status other then success
#
# implementation class
filter.1.class=com.sterlingcommerce.woodstock.alert.filter.XPathFilter
# filter name (must be unique)
filter.1.name=xpathfilter_bp_status_notSuccess
# Filter's criteria
(in this case we ignore BProcess of status "SUCCESS" or
        "WAITING" and state "ACTIVE" )
filter.1.criteria=//EVENT/@type="BUSINESS_PROCESS" and
not(//wf_status=0 or //wf_status=3) and  not(//wf_state=0)
#
# Filter for events type "ALERTER"
#
filter.2.class=com.sterlingcommerce.woodstock.alert.filter.XPathFilter
filter.2.name=xpathfilter_type_alerter
filter.2.criteria=//EVENT/@type="ALERTER"
```

### Implementing New Filter Classes to Plug In

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

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

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

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

## Implementing New Alerter Classes to Plug In

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

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

# Chapter 3. Archive Business Process Service

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

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

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

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

## How the Archive Business Process Service Works

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

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

to begin archiving data according to the parameters you have specified.

# Chapter 4. Attachment Parsing Service

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

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

## Business Scenario

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

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

2. Configure two business processes: SMTPExcelDoc and Attachment_Parsing_Process.

## SMTPExcelDoc Business Process Example

The following example illustrates the SMTPExcelDoc business process:

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

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

## Attachment_Parsing_Process Business Process Example

The following example illustrates the Attachment_Parsing_Process business process:

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

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

## Output from Business Process to Service

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

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

## Implementing the Attachment Parsing Service

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

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

# Chapter 5. Auto Terminate Service

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

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

## How the Auto Terminate Service Works

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

## Implementing the Auto Terminate Service

A pre-configured instance of the Auto Terminate service is provided with Sterling Integrator and may be used in your business processes. You may also create new instances.

To implement a new instance of the Auto Terminate service, complete the following tasks:

1. Create a new configuration of the Auto Terminate service. For basic information about creating service configurations, see *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Auto Terminate Service*.

2. Specify field settings for the service configuration in the Sterling Integrator Admin Console as necessary.

## Configuring the Auto Terminate Service

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

## Creating a Service Configuration in the Admin Console

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

| Field | Description |
|---|---|
| Name | Unique and meaningful name for the service configuration. Required. |
| Description | Meaningful description for the service configuration, for reference purposes. Required. |
| Select a Group | Select one of the options:<br>• None – Do not include the configuration in a service group at this time.<br>• 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.)<br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list.<br><br>**Note:** For more information about service groups, see *Managing Services and Adapters*. |
| Run As User | Type the user ID to associate with the schedule, or click the<br><br>icon and select a user ID from the list. Valid values: Any valid Sterling Integrator user ID. |
| Use 24 Hour Clock Display | Check the box to use the 24-hour clock instead of the default 12-hour clock. |

| Field | Description |
|---|---|
| Schedule | Specify the scheduling information for running the Auto Terminate service. Valid values: |
| | • Do not use schedule If you select this field, the service does not run on a schedule. |
| | • Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the service to run at startup. |
| | • Run daily Valid values are the hour and minutes at which to run the service each day. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. |
| | • Run based on days 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 days of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |

### Setting Up the Service in the GPM

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

### Overriding the bprecovery.properties File Settings

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

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

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

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

1. In the *install_dir*/properties directory, locate (or create, if necessary) the customer_overrides.properties file.
2. Open the customer_overrides.properties file in a text editor.
3. Add the properties that you want to override, using the following format:

bprecovery.*PROPERTY_NAME=PROPERTY_VALUE*

| Property | Description |
|---|---|
| auto_terminate_days | Number of days that a business process must be in one of the specified states before being terminated by the Auto Terminate service. Valid values: any positive integer. Default is 14. |
| num_states | Number of different specified business process states to use as a condition for terminating a business process. Must correspond to the number of auto_terminate_statex properties. For example, if there are 3 **auto_terminate_statex** properties, **num_states** must be set to 3. Default is 1. |
| auto_terminate_statex | Business process state to use as a condition for terminating a business process. $x$ = a digit corresponding to the number of the state. For example, the first state would be auto_terminate_state1.Default is auto_terminate_state1=halted.The number of **auto_terminate_statex** statements must correspond to the value of **num_states**. For example, if **num_states** is set to 3, there must be 3 **auto_terminate_statex** properties: <br>• auto_terminate_state1=*1st state* <br>• auto_terminate_state2=*2nd state* <br>• auto_terminate_state3=*3rd state* <br><br>Valid states are: <br>• interrupted_man <br>• interrupted_auto <br>• halted <br>• waiting |
| auto_terminate_batch | Maximum number of business processes that will be marked during each run of the Schedule_AutoTerminateService business process. Default is 1000. |

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

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

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

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

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

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

The new value can be one of the following:

- double – Doubles the value of **auto_terminate_days** in the bprecovery.properties file.
- an integer value – A new integer value. Partial days can also be defined here. For example, 0.5 would specify one-half of a day.

## Business Process Examples

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

```
Schedule_AutoTerminateService
<process name="Schedule_AutoTerminateService">
  <rule name="Exist-PDOC">
    <condition>count(//PrimaryDocument) &gt; 0</condition>
  </rule>
  <rule name="Exist-PDOC-onfault">
    <condition>count(//PrimaryDocument) > 0</condition>
  </rule>
 <sequence>
    <operation name="SetLock">
     <participant name="SystemLockService"/>
     <output message="Xout">
        <assign to="LOCK_KEY">AutoTerminateHaltedBPS</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="CLEAR_ON_START_UP">true</assign>
        <assign to="." from="*"></assign>
     </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
     </operation>
   <operation name="Service">
      <participant name="AutoTerminateService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
      </output>
     <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="unLock">
       <participant name="SystemLockService"/>
       <output message="Xout">
        <assign to="ACTION">unlock</assign>
        <assign to="LOCK_KEY">AutoTerminateHaltedBPS</assign>
       <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
     </operation>
   <sequence name="email">
      <choice name="exist pdoc">
        <select>
         <case ref="Exist-PDOC" activity="sending-email"/>
```

```
                    </select>
                    <sequence name="sending-email">
                      <operation name="Get mailfrom">
                        <participant name="UserService"/>
                        <output message="UserServiceTypeInputMessage">
                          <assign to="MethodName">getEmail</assign>
                          <assign to="UserName">admin</assign>
                          <assign to="." from="*"></assign>
                        </output>
                         <input message="inmsg">
                         <assign to="mailfrom" from="//UserService/getEmail/text()">
</assign>
                        </input>
                       </operation>
                      <operation name="get mailto">
                        <participant name="UserService"/>
                        <output message="UserServiceTypeInputMessage">
                          <assign to="MethodName">getEmail</assign>
                          <assign to="UserName">admin</assign>
                          <assign to="." from="*"></assign>
                        </output>
                         <input message="inmsg">
                         <assign to="mailto" from="//UserService/getEmail/text()">
</assign>
                        </input>
                       </operation>
                      <operation name="SMTP Send">
                        <participant name="SMTP_SEND_ADAPTER"/>
                        <output message="SMTP_SEND_ADAPTERInputMessage">
                          <assign to="xport-smtp-mailfrom" from="//mailfrom/text()">
</assign>
                          <assign to="xport-smtp-mailhost" from="'00.000.000.00'">
</assign>
                          <assign to="xport-smtp-mailport">25</assign>
                          <assign to="xport-smtp-mailto" from="//mailto/text()"></assign>
                          <assign to="xport-smtp-mailsubject">Automated Terminate Notification
                          </assign>
                          <assign to="." from="*"></assign>
                        </output>
                         <input message="inmsg">
                           <assign to="." from="*"></assign>
                        </input>
                      </operation>
                    </sequence>
                  </choice>
               </sequence>
                <onFault>
                 <sequence>
                    <operation>
                     <participant name="SystemLockService"/>
                     <output message="Xout">
                       <assign to="ACTION">unlock</assign>
                       <assign to="LOCK_KEY">AutoTerminateHaltedBPS</assign>
                       <assign to="." from="*"></assign>
                     </output>
                      <input message="Xin">
                      <assign to="." from="*"></assign>
                     </input>
                    </operation>
                  <sequence name="email">
                    <choice name="exist pdoc">
                      <select>
                         <case ref="Exist-PDOC-onfault"
activity="sending-email-onfault"/>
                      </select>
                      <sequence name="sending-email-onfault">
                        <operation name="Get mailfrom">
```

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

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

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

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

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

# Chapter 6. B2B Lookup System Service

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

| System name | B2B_LOOKUP_SYSTEM |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, System |
| Description | Part of the system business process B2B_LOOKUP. This service always runs by the user services B2B_SEND, B2B_RESPOND, B2B_REQUEST_RESPONSE. The purpose of this service is to look up the trading profile information in the contract. This service takes the b2b-contract-id and b2b-message-mode that are in the business process context and searches the appropriate b2b-profile-id and sets that in the business process context. This service also looks for the appropriate transport business process for the transport associated with the contract. If the b2b-message-mode is: <br><br>• Respond, the service uses the Production profile <br><br>• Send, the service uses the Consumption profile <br><br>• Request-response, the service uses the Production profile |
| Business usage | This service is not designed to be used as part of a business process. It is used by other services within Sterling Integrator. |
| Usage example | None |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | B2B Send, B2B Response, and B2B Request Response. |
| Application requirements | No |
| Initiates business processes? | No |
| Invocation | Run only by other services instead of a business process. |
| Business process context considerations | No |
| Returned status values | Name: B2B_LOOKUP Instance ID: *35* Service Name: B2B Lookup System service Contract ID=*6fc193:e837daefd6:-7fc0* Profile ID=*6fc193:e837daefd6:-7ff9* Transport Workflow Name=HTTP_SEND_ENVELOPE_OFF Transport Type=HTTP Message Mode=send |
| Restrictions | The trading profile information and contract must be configured before using this service. The service expects two parameters in the business process context: <br><br>• b2b-contract-id <br><br>• b2b-message-mode <br><br>There should be only one instance of this service. It is a system service. It should not be used in any user-created business processes. |

| | |
|---|---|
| Persistence level | None |
| Testing considerations | This service is tested in conjunction with the following services: B2B Send, B2B Response, and B2B Request Response. |
| Notes | None |

# Chapter 7. B2B Mail Client Adapter

The B2B Mail Client adapter enables Sterling Integrator to:

- Collect mail from an RFC 1725-compliant mail server using the POP3 protocol or an RFC 2060-compliant mail server using the IMAP protocol.
- Locate an appropriate business process or contract.
- Start a business process with the data appended.

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

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

| | |
|---|---|
| System name | TEST_MAIL_CLIENT_ADAPTER |
| Graphical Process Modeler (GPM) categories | All Services, System |
| Description | Collects mail from an RFC 821-compliant mail server using the POP3 or IMAP protocol. The adapter can start a business process with the mail data it retrieves from the mail server. It can also place all the mail headers in the business process context for further processing. |
| Business usage | Polls the mail host at a specific interval to retrieve any mail in the mailbox. If there is no mail, the adapter tries again at the next interval. |
| Usage example | Some customers in remote locations send EDI data into a mail server mailbox through SMTP. A business process can be built to collect all new EDI data from the mailbox, and forward the results to the translation services. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | No |
| Application requirements | Requires an RFC 821-compliant mail server and a mail server that supports POP3 or IMAP. The Mail server should be configured to mark mail as SEEN, ANSWERED, NEW, or DELETED when they are requested from B2B Mail Client adapter. |
| Initiates business processes? | Yes |
| Invocation | Not applicable (internal service) |
| Business process context considerations | This adapter can be configured to extract mail header information and place it into process data. |
| Returned status values | None |
| Restrictions | This adapter will connect to a mail server that supports POP3 or IMAP. The mail server should be RFC 821. |
| Persistence level | System default |
| Testing considerations | None |

## How the B2B Mail Client Adapter Works

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

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

## Process Data structure

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

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

## Process Data Example

The following example shows how the process data might look:

```
<ProcessData>
<PrimaryDocument SCIObjectID="server:20187f:ef9c723ba3:-7fa4"/>
<Mail_Client>
<Headers>
<From>"Boston, Test1" <test1@mycompany.com></From>
<Subject>xml file</Subject>
<To>"Boston, Test2" <test2@mycompany.com>, "Boston, Test3"
  <test2@mycompany.com></To>
<Date>Mon, 12 Aug 2002 10:48:28 +0800</Date>
<Received>by aaasgpxs00.boston.mycompany.comid
      <01C241AA.BCF67770@aaasgpxs00.boston.mycompany.com>;
       Mon,12 Aug 2002 10:48:29 +0800</Received>
<MIME-Version>1.0</MIME-Version>
<Content-Type>application/octet-stream;
 name="smtpSetSubject.xml"</Content-Type>
<Content-Disposition>attachment;
 filename="smtpSetSubject.xml"</Content-Disposition>
<Message-ID><EA43E744516FD311A2FB00903B5C499CFF9B64@aaasgpxs00.
            boston.mycompany.com>
     </Message-ID>
</Headers>
</Mail_Client>
</ProcessData>
```

## Initiating a Business Process that Contains the B2B Mail Client Adapter

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

the business process ID is obtained from the contract and the adapter starts that business process. If a business process match is found, the adapter starts that business process.

Configuring Multiple URIs

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

**Business Process Example**

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

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

## Implementing the B2B Mail Client Adapter

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

1. Create a B2B Mail Client adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the B2B Mail Client adapter. For information, see *Configuring the B2B Mail Client Adapter*.

   **Note:** When configuring the B2B Mail Client adapter, consider that you cannot pass parameters to the B2B Mail Client adapter from BPML or configure the adapter using the GPM.
3. Use the B2B Mail Client adapter in a business process. For information, see *Initiating a Business Process that Contains the B2B Mail Client Adapter* .

## Configuring the B2B Mail Client Adapter

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

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

| Field | Description |
|---|---|
| Description | Meaningful description for this adapter configuration, for reference purposes. Required. |
| Select a Group | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• Select Group – If you have already created one or more groups for this adapter type, they are displayed in the list. Select a group from the list. |
| Default Message Access Protocol | Select the mail accessing protocol: POP3 or IMAP. Required. |
| Default Mail Server | Name of the mail server that Sterling Integrator is to connect to. This is any valid DNS name or a dotted decimal address. |
| Default Mail Server Port | Port number that is monitored for activity by the remote system. This must be a valid port and must be 1 - 5 numeric characters. |
| User Name | Name that identifies the mailbox to provide when connecting to a mail server. This field is case-sensitive. |
| Password | Password used for authentication when connecting to a mail server. This field is case-sensitive. |
| Connection Retries | The number of time Mail Client adapter tries to connect to the mail server when there is a connection error. Optional. Minimum valid value is 1 (default is 3). |
| Retry Interval (Sec) | The retry interval specifies in seconds. Optional. Minimum valid value is 1 (also the default). |
| Max Messages per Session | Maximum number of messages that can be processed per mail download session. Optional. Minimum valid value is 1 (default is 500). |
| Remove inbox mail messages | To remove the mail message from the inbox after successfully downloading them. Only configurable if using IMAP protocol. If POP3 is used to access the mailbox, the messages will be removed from the inbox after they have been successfully downloaded. Optional. Valid values are Yes and No. |
| SMIME Encryption User Certificate | Not used in this release. Leave at default. |
| SMIME Decryption User Certificate (System Store) | Not used in this release. Leave at default. |
| SSL | Whether to use SSL to connect to mail server. Optional. Valid values are Must and None. If Must is selected, you must configure the SSL parameters that follow Key Certificate Passphrase. |
| Key Certificate Passphrase | The passphrase for the key certificate. Optional. |
| Cipher Strength | The cipher strength use for SSL. Optional. Valid values are Strong, Weak, and All. |
| Key Certificate (System Store) | Object ID that contains the privatekey and public certificate. The trading partner trusted certificate authority signs the Certificate. Optional. |

| Field | Description |
|---|---|
| CA Certificates | Object ID that contains the trusted certificate authority public certificate. Optional. |
| Add New URI | Click **add** to add a new Uniform Resource Indicator (URI). |
| **Note:** The following four fields are displayed only if **Add New URI** is selected: | |
| URI | Uniform Resource Indicator. Use the same name that is provided in the User Name field above. This field is case-sensitive and is 1 - 8 alphanumeric characters beginning with a letter. |
| Business Process | Name of the business process you want the B2B Mail Client adapter to start. This field is valid only if no contract is defined. Either a contract or a business process can be associated with this adapter, but not both. |
| Contract | Name of the contract to associate with this adapter. This is valid if no business process is defined. Either a contract or a business process can be associated with this adapter, but not both. |
| Send Raw Messages | Whether to send raw messages that preserve the mail headers in the body of the message. Valid values are Yes and No. |
| Insert into Process Data | Whether to extract all the mail headers information from the mail message and put them into process data. Required. Valid values are All mail header fields and Selected mail header fields (default is All mail header fields). If Selected mail header fields is chosen, Sterling Integrator will only attempt to extract the values you choose from the following four pages. |
| Mail Header: Originator Fields | Only displayed if **Selected mail header fields** was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: From, Resent-from, Sender, Resent-Sender, Reply-to, Resent-reply-to. Optional. |
| Mail Header: Receiver Fields | Only displayed if **Selected mail header fields** was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: To, Resent-to, CC, Resent-cc, BCC, Resent-bcc. Optional. |
| Mail Header: Reference Fields | Only displayed if **Selected mail header fields** was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: Message-id, Resent-message-id, In-Reply-to, References, Keywords. Optional. |
| Mail Header: Other Fields | Only displayed if **Selected mail header fields** was selected for Insert into Process Data parameter. Select one or more of the following fields to insert into process data: Subject, Comments, Encrypted, Date, Resent-date, Attachments. Optional. |
| Run as User | Enter (or select from the list) the user ID to be associated with business process instances of this service. |
| Use 24 Hour Clock Display | Select to specify times for this schedule using the 24 hour clock. Leave blank to use 12 hour clock and AM and PM. |

| Field | Description |
|---|---|
| Schedule | Information about scheduling the adapter configuration to run and to start the specified business process. Valid values:<br><br>• Do not use schedule<br><br>If this field is selected, this service does not start a business process and does not run on a schedule.<br><br>• Run based on timer<br><br>Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup.<br><br>• Run daily<br><br>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.<br><br>• Run based on days of the week<br><br>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.<br><br>• Run based on days of the month<br><br>Valid values are the day of the month (including the last day of the month (LDOM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. |

# Chapter 8. B2B Request/Response System Service

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

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

# Chapter 9. B2B Send System Service

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

| System name | B2B_SEND |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | The B2B Send system service sends messages to trading partners. It does this by retrieving and using information from trading profiles and the associated trading profile contract to identify how and where to send information. |
| Business usage | Use this service in a business process to send a document or message to a trading partner identified as the consumption profile (receiving partner) in a contract. |
| Usage example | Send messages to a trading partner. |
| Preconfigured? | Yes |
| Requires third-party files? | Yes |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | This service is one of four services that are used in Sterling Integrator for B2B communications using trading profiles. The other three services are:<br><br>• B2B Respond service – Responds to messages from a production (sending) trading partner using trading profile information in the contract.<br><br>• B2B Request Response service – Uses the production (sending) profile of a contract to send a request to and retrieve a response from the production trading partner.<br><br>• B2B Lookup service – Finds trading partner contract records and retrieves profile information. It is a system service included automatically in your business process if you include the B2B Respond service, the B2B Request Response service, or the B2B Send service.<br><br>Two other services provide EDI documents that can be sent using the B2B Send adapter. These services are:<br><br>• EDI Encoder service – Determines which enveloping services need to run and sends the document to the translator for translation, if needed.<br><br>• EDI Enveloping service – Determines which business processes need to run to apply EDI envelopes.<br><br>**Note:** The EDI Encoder and Enveloping services prepare the document to be sent. |

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

## About B2B Services and Trading Partner Setup

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

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

## How the B2B Send Service Works

The B2B Send service is preconfigured and installed as part of Sterling Integrator. No additional configuration of this service is necessary.

When using the B2B Send service in a business process:
1. The service runs as part of the business process.
2. This service starts the B2B Lookup service.
3. The B2B Lookup service locates:
    - The consumption profile of the contract

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

    The default transport business processes are:

    HTTP_SEND_ENVELOPE_OFF

    FTP_SEND_ENVELOPE_OFF

    SMTP_SEND_ENVELOPE_OFF

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

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

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

## Business Scenario

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

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

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

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

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

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

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

## GPM Configuration for the EDI_Outbound_To_TP Business Process Example
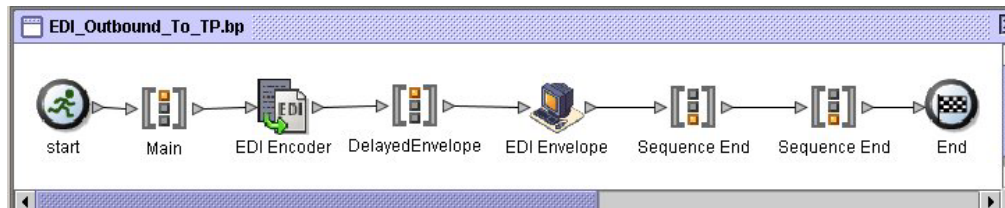


1. The EDI_Outbound_To_TP business process receives the XML file as the primary document.
2. The EDI Encoder service takes the sender, receiver, and lookup alias as input parameters. Based on this information, it determines which enveloping services to run and sends the document to the translator, if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

## GPM Configuration for the SendOutBoundDocuments Business Process Example
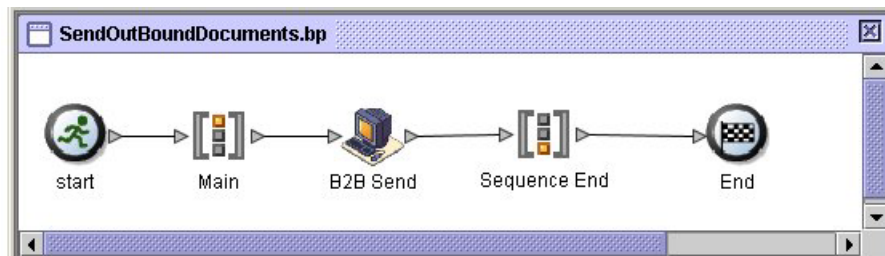


The SendOutBoundDocuments Business Process consists only of the B2B Send service. In this case, it always runs as a subprocess and the b2b-contract-id is in process data when it is run. There is no additional configuration required for this service in the GPM.

## Process Data Example

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

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

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

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

## Contract Configuration with the SendOutBoundDocuments Business Process Example

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



## Business Process Modeling Language (BPML) Example

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

• EDI_Outbound_To_TP Business Process

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

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

- SendOutBoundDocuments Business Process

```
<process name="SendOutBoundDocuments">
 <sequence name="Main">
 <operation name="B2B Send">
  <participant name="B2B_SEND"/>
  <output message="B2B_SEND_SERVICEInputMessage">
  <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
  <assign to="." from="*"></assign>
  </input>
 </operation>
 </sequence>
</process>
```
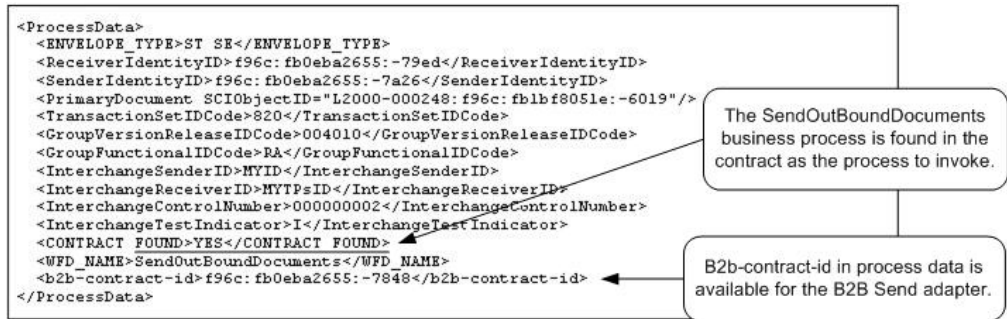
The Business Process Monitor shows the order in which the business processes ran.

| Status | ID | Name | State | Started | Ended | Expires | Parent/Child |
|--------|-----|------|-------|---------|-------|---------|--------------|
| 🗸 | 43518 | HTTP_SEND_ENVELOPE_OFF | Active | 3/11/04 1:30:01 PM | | | △ |
| 🗸 | 43517 | B2B_LOOKUP | Waiting | 3/11/04 1:30:01 PM | | | △ ▽ |
| 🗸 | 43516 | SendOutBoundDocuments | Waiting | 3/11/04 1:30:01 PM | | | △ ▽ |
| 🗸 | 43515 | X12EnvelopeUnified | Completed | 3/11/04 1:29:57 PM | 3/11/04 1:30:01 PM | | △ ▽ |
| 🗸 | 43514 | EDI_Outbound_ToTP.bp | Completed | 3/11/04 1:29:56 PM | 3/11/04 1:29:58 PM | | ▽ |

## Implementing the B2B Send Service

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

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

1. Configure the trading profile and a corresponding contract and obtain the b2b-contract-id from the trading profile contract.

2. Use the B2B Send service in a business process.

## Editing the B2B Send Service

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



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

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

| Parameter Name | Description |
| --- | --- |
| Config | Name of the Service configuration. Required. |
| b2b-contract-id | The contract ID with the transport information to be used to send the message. This value is input from the business process in process data. Valid value is any valid contract ID. |
| b2b-raw-response | Whether to receive a raw response. Optional. Valid values are:<br>• true - receive raw response<br>• false - do not receive raw response (Default)<br>Can be used interchangeably with xport-raw-response. |
| xport-raw-response | Whether to receive a raw response. Optional. Valid values are:<br>• true - receive raw response<br>• false - do not receive raw response (Default)<br>Can be used interchangeably with b2b-raw-response. |

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

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

## Usage Example

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

## GPM Example

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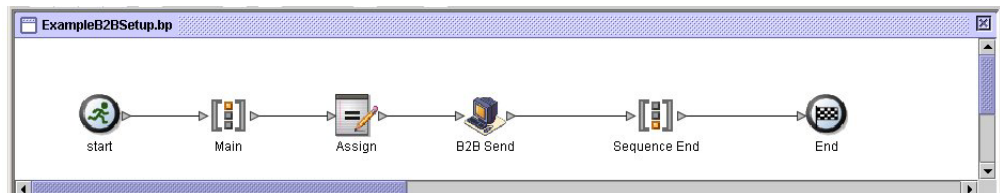


The following figure shows the graphical view of the GPM parameters for the Assign activity. The contract ID (3c9217:f5894f9-24ca) is assigned to b2b-contract-id. This is a required parameter for the B2B Send service and is found in the trading partner contract information. The B2B Lookup service looks up the trading partner transport information and uses the specified transport protocol (such as HTTP, SMTP, or FTP) to send the message.



## BPML Example

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

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

```
    </input>
  </operation>
  </sequence>
</process>
```

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

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

## Example Using b2b-raw-response

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

# Chapter 10. B2B SMTP Client Adapter

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

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

# Chapter 11. Backup Business Process Service

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

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

## How the Backup Business Process Service Works

The Backup Business Process service is invoked by the BackupService business process based on settings in the Archive Configuration section of the Archive Manager. By default, the Backup Business Process service will run daily at 2:00 a.m. Business processes specified in the Archive Manager to be archived will be saved in a backup set. The Backup Business Process service backs up the specified business process data and removes it from the database.

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

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

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

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

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

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

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

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

## Implementing the Backup Business Process Service

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

# Configuring the Backup Business Process Service

You can specify field settings in 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 | Select one of the options:<br>• None – Do not include the configuration in a service group at this time.<br>• 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.)<br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list.<br><br>**Note:** For more information about service groups, see *Managing Services and Adapters.* |
| Thread Pool Size | Number of execution threads allocated to archiving. Required. Default is 8.<br>**Note:** Although allocating more threads usually increases performance, never allocate more threads than the number of tables plus one. Allocating too many threads can decrease performance. |
| Number of Business Processes per Backup Set Chunk | Maximum number of business processes grouped together within a single chunk of a backup set. Required. Default is 10,000. |
| Maximum Number of Chunks per Invocation | Maximum number of chunks that will be backed up in a single backup set. Required. Default is 25. |
| Compress Data | Whether to compress the data before archiving. Valid values:<br>• true – Compress the data. This reduces the amount of disk space required for storing the backup set.<br>• false – Do not compress the data. This improves backup speed and performance at the expense of disk space. |
| Maximum Backup File Size | Maximum size of a backup file before creating a new file. Required. Default is 1. |
| Archive File Size Unit of Measure | Unit of measure used for **Maximum Backup File Size**. Required. Valid values:<br>• K – Kilobyte<br>• M – Megabyte<br>• G – (Default) Gigabyte<br>• T – Terabyte |
| Maximum Number of Days per Backup Set | Maximum number of days that will be backed up in one backup set. Required. Default is 2. |

| Field | Description |
|---|---|
| Run as User | 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.<br>**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. |
| Use 24 Hour Clock Display | If selected, the service will use the 24-hour clock instead of the default 12-hour clock. |
| Schedule | Information about scheduling the business process invoked by the Backup Business Process service. Valid values:<br>• Do not use schedule If this field is selected, the service does not run on a schedule.<br>• 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.<br>• 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.<br>• Run based on days 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.<br>• Run based on days 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. |

# Chapter 12. Batch Processor Service

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

| System name | BatchProcessorService |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | The Batch Processor service is used to merge XML or non-XML documents or document fragments. If specified, XML documents can be translated through a map before being merged. In non-XML processing, the documents (prefix, suffix, and documents referenced by the input document) are treated as binary files and concatenated. The use of document streaming in the Batch Processor service allows the processing of large documents, allowing fast and efficient merges. |
| Business usage | Use the Batch Processor service in situations such as the following: <br>• You want to create one document out of multiple XML documents, but only fragments of the documents need to be merged and some values need to be replaced. <br>• You have a need to pick up and concatenate a group of non-XML documents during EDI-SAP integration processing. |
| Usage example | Fragments of documents need to be combined into one document, and some values need to be replaced. The Batch Processor sends each document through a map, which replaces some of the values in the document. Then a fragment of each document is parsed out and added to an output document. Each document goes through the same process. Once all documents have been processed and added to the output document, the output document is placed in process data. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | No |
| Application requirements | No |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values: <br>• Success – The service successfully created an output document <br>• Error – The service was not successful in creating an output document. |
| 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 | None |

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

## Implementing the Batch Processor Service

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

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

## Configuring the Batch Processor Service

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| prefix_document(prefix_document) | The content of the XML document referenced will be added to the beginning of the output document. If a document does not exist with the value specified, XPath will be applied to the value and the text retrieved by xpath added to the beginning of the document (xpath and text needs to be placed in ProcessData). If the xpath does not exist either, nothing will be added. The value has to match the name of a document or xpath. If neither exists, nothing will be added to the beginning of the output document. Optional. |
| suffix_document(suffix_document) | The content of the XML document referenced will be added to the end of the output document. If a document does not exist with the value specified, xpath will be applied to the value and the text retrieved by xpath added to the end of the document (xpath and text needs to be placed in ProcessData). If the xpath does not exist either, nothing will be added. The value has to match the name of a document or xpath. If neither exists, nothing will be added to the beginning of the output document. Optional. |
| input_document(input_document) | This document contains references to other documents that will be added between the prefix_document and suffix_document. In case of XML documents, each one referenced might be partially added, translated using a map, and/or have values replaced (non-XML documents are concatenated without additional processing. The value has to match the name of a document. Optional. |
| use_xml | Indicates XML or non-XML processing. Valid values: true (default), false. If false (non-XML processing), the files collected are treated as binary data and concatenated, with no further processing. Optional. If nothing is specified for this field when a business process is created, the default is used. |

## Business Process (BPML) Example: XML Processing

The following BPML runs the Batch Processor service. Three wfd parameters are defined: prefix_document, suffix_document, and input_document:

```
<process name="default">
 <operation name="BatchProcessor">
  <participant name="BatchProcessorService"/>
  <output message="BatchProcessorInputMessage">
   <assign to="input_document">InputDocument</assign>
   <assign to="prefix_document">PrefixDocument</assign>
   <assign to="suffix_document">SuffixDocument</assign>
   <assign to="use_xml">true</assign>
   <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
   <assign to="." from="*"></assign>
  </input>
 </operation>
</process>
```

In this example, consider that:

- The prefix_document name is PrefixDocument and suffix_document name is SuffixDocument. If the PrefixDocument exists, content of this document will be added to the beginning of the output document.
- If the PrefixDocument does not exist, the Batch Processor Service will look in ProcessData and apply xpath to PrefixDocument. If /ProcessData/ PrefixDocument element exists, the value for that element will be added to the beginning of the output document.
- If the PrefixDocument does not exist, nor does /ProcessData/PrefixDocument element, nothing will be added to the beginning of the output document.
- The same processing rules that apply to PrefixDocument also apply to SuffixDocument, except that the SuffixDocument will be added to the end of the output document
- The input_document name is InputDocument. If InputDocument does not exist, an error will be thrown. The InputDocument contains references to other documents that enables the service to pass the output to the PrimaryDocument.
- The field use_xml indicates that XML processing is used.

The following example show a PrefixDocument that is added to the beginning of the output document:

```
<BatchProcessorOutput>
```

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

```
</BatchProcessorOutput>
```

The following example shows an InputDocument:

```
<batchProcessor>
<fragment docname="Doc1" mapname="BatchProcessorMap1">
        <replaceList>
            <value>valueoffirstiteminmap1</value>
            <value>valueoffseconditeminmap1</value>
            <value>valueoffthirditeminmap1</value>
        </replaceList>
        <replaceList>
            <value>valueoffirstiteminmap2</value>
            <value>valueoffseconditeminmap2</value>
            <value>valueoffthirditeminmap2</value>
```

```
            </replaceList>
        </fragment>
        <fragment docname=" Doc2" mapname="BatchProcessorMap2"
              xpath=" /example1/element1/batch">
            <replaceList>
                <value>valueoffirstiteminmap3</value>
                <value>valueoffseconditeminmap3</value>
            </replaceList>
        </fragment>
</batchProcessor>
```

For this example, consider that:

- This document has to be structured as the example shown below with batch processor and fragment elements.
- The fragment element has three attributes: docname, mapname, xpath. The docname is mandatory, whereas xpath and mapname are optional.
- The docname is the name of the document whose content will be added to the output document.
- The mapname is the name of the map that will translate the document.
- The xpath is applied to the document once it has been translated, and will parse out a document fragment. This document fragment will then be added to the output document.
- If value elements exist under the replaceList element, these are used for replacing values in the map. Behind the scenes, these values will be placed in ProcessData as batchProcessingVar1, batchProcessingVar2, etc. The map can then (if desired) pick up these values and replace values during translation of a document. To add this rule to a map, go to Standard Rule. Under "Please enter the Xpath to evaluate", enter batchProcessingVar1/text() etc.
- Doc1 is added to the output document twice, but with different replacement values. Doc2 will then be added once, but only a fragment of the document will be added (since xpath is specified).

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

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

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

```
<example2>
    <element1>
        <batch>
```

```
            <value>111</value>
            <value>222</value>
        </batch>
        <tag1>tag1</tag1>
        <tag2>tag2</tag2>
        <tag3>tag3</tag3>
    </element1>
</example2>
```

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

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

## Business Process (BPML) Example: Non-XML Processing

In this example, the assign statement "use_xml=false" indicates non-XML processing. The files collected are treated as binary data and concatenated, with no further processing.

```
<process name="BatchProcessor">
  <sequence>
    <operation name="File System Adapter">
      <participant name="FSA"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="attachFile">/server/share/shared_directory/pgp/prefix.xml
                </assign>
        <assign to="deleteAfterCollect">false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <assign to="PrefixDocument" from="//PrimaryDocument/@SCIObjectID">
</assign>
    <operation name="File System Adapter">
```

```
          <participant name="FSA"/>
          <output message="FileSystemInputMessage">
      <assign to="Action">FS_COLLECT</assign>
         <assign to="attachFile">/server/share/shared_directory/pgp/suffix.xml
            </assign>
         <assign to="deleteAfterCollect">false</assign>
         <assign to="." from="*"></assign>
       </output>
       <input message="inmsg">
         <assign to="." from="*"></assign>
       </input>
      </operation>
      <assign to="SuffixDocument" from="//PrimaryDocument/@SCIObjectID">
</assign>
     <operation name="File System Adapter">
       <participant name="FSA"/>
       <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
         <assign to="attachFile">/server/share/shared_directory/pgp/input.xml
              </assign>
         <assign to="deleteAfterCollect">false</assign>
         <assign to="." from="*"></assign>
       </output>
       <input message="inmsg">
         <assign to="." from="*"></assign>
       </input>
      </operation>
      <assign to="InputDocument" from="//PrimaryDocument/@SCIObjectID">
</assign>
     <operation name="File System Adapter">
          <participant name="FSA"/>
          <output message="FileSystemInputMessage">
            <assign to="Action">FS_COLLECT</assign>
            <assign to="attachFile">
/server/share/shared_directory/pgp/Doc1.xml
                </assign>
            <assign to="deleteAfterCollect">false</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>
      <assign to="Doc1.xml" from="//PrimaryDocument/@SCIObjectID">
</assign>
     <operation name="File System Adapter">
          <participant name="FSA"/>
             <output message="FileSystemInputMessage">
               <assign to="Action">FS_COLLECT</assign>
               <assign to="attachFile">
/server/share/shared_directory/pgp/Doc2.xml
                   </assign>
               <assign to="deleteAfterCollect">false</assign>
               <assign to="." from="*"></assign>
             </output>
             <input message="inmsg">
               <assign to="." from="*"></assign>
           </input>
           </operation>
     <assign to="Doc2.xml" from="//PrimaryDocument/@SCIObjectID"></assign>
     <operation name="BatchProcessor">
       <participant name="BatchProcessorService"/>
       <output message="BatchProcessorInputMessage">
         <assign to="input_document">InputDocument</assign>
         <assign to="prefix_document">PrefixDocument</assign>
         <assign to="suffix_document">SuffixDocument</assign>
   <assign to="use_xml">false</assign>
```

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

# Chapter 13. BEA Tuxedo Adapters

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

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

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

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

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

## How the BEA Tuxedo Adapters Work

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

The following figure shows how Sterling Integrator interacts with a BEA Tuxedo system:
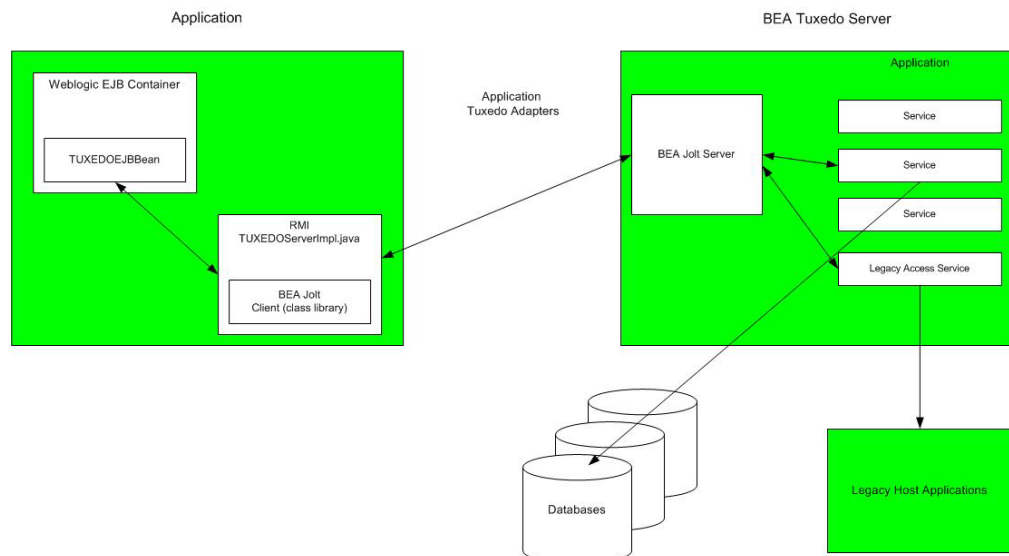
The following steps summarize how the Tuxedo adapters work within a business process in Sterling Integrator:

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

## Tuxedo Subscribe Adapter

Using the BEA Jolt client class library (jolt.jar file), Sterling Integrator can support subscriptions to Tuxedo event-based notifications. When subscribing to event notifications, BEA Jolt internally enables the notification event handler for each event session—thus generating and sending notifications for both unsolicited and brokered events. Using the Tuxedo Subscribe adapter, you can override the event handler process for the event session. That is, you can disregard the onReply() method in the event handler process and define what should happen to all notifications run for the client for that session.

Sterling Integrator supports the following types of event-based notifications:

- Unsolicited Event – Generates and sends notifications for all business events, including business events that are broadcast or posted.
- Brokered Event – Generates and sends notifications for only specified published business events.

## Tuxedo Application Services Adapter

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

## Implementing the BEA Tuxedo Adapters

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

1. Activate your license for the BEA Tuxedo adapters. See *An Overview of Implementing Services*.
2. Install the adapters. See *Installing the BEA Tuxedo Adapters*.

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

3. If subscribing to event-based notifications, create a Tuxedo Subscribe adapter configuration to generate notifications. See *Creating a Service Configuration* and *Configuring the BEA Tuxedo Adapters*.

4. Create a configuration of the Tuxedo Application Services adapter. Configure the adapter to retrieve an application and its services. See *Creating a Service Configuration* and *Application BEA Tuxedo Application Services Adapter Configuration*.

5. Use the Tuxedo Application Services adapter in a business process.

   See *Creating a Business Process Using the BEA Tuxedo Application Services Adapter* to define fields that complete request-and-response transactions for Tuxedo adapters.

## Installing the BEA Tuxedo Adapters

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

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

To install the Tuxedo adapters:
1. Verify that BEA Jolt is installed on the computer where the BEA Tuxedo system is installed.

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

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

   Path: _____

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

   Path: _____

4. Shut down Sterling Integrator if it is running.

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

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

6. Restart Sterling Integrator.

## Configuring the BEA Tuxedo Adapters

To configure the Tuxedo adapters, you must specify field settings in Sterling Integrator and in the GPM. That is:
- Configure the Tuxedo Subscribe adapter to subscribe to Tuxedo event-based notifications.

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

## Sterling Integrator BEA Tuxedo Subscribe Adapter Configuration

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

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

## Sterling Integrator BEA Tuxedo Application Services Adapter Configuration

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

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

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

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

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

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

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

- Create another configuration of the Tuxedo Application Services adapter to retrieve another application and services from the BEA Tuxedo system.
- Create a business process using the configuration of the Tuxedo Application Services adapter that you just created.

## Creating a Business Process Using the BEA Tuxedo Application Services Adapter

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

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

- An icon for each adapter configuration that you create using the Tuxedo Application Services adapter displays in the Custom Services stencil.
- To view adapter properties, you must drag the icon that represents an adapter configuration to the workspace. Double-click the icon, and select the configuration from the Config list in the Service Editor.
- The Action list displays the services request-and-response transactions for business process use.

The following figure shows the GPM. The BEA Tuxedo Application Service adapter has been configured and renamed to Tuxedo BankApp, and an adapter configuration called BankApplication is selected. The Action parameter displays a list of the services specified during configuration and retrieved from a BEA Tuxedo system.
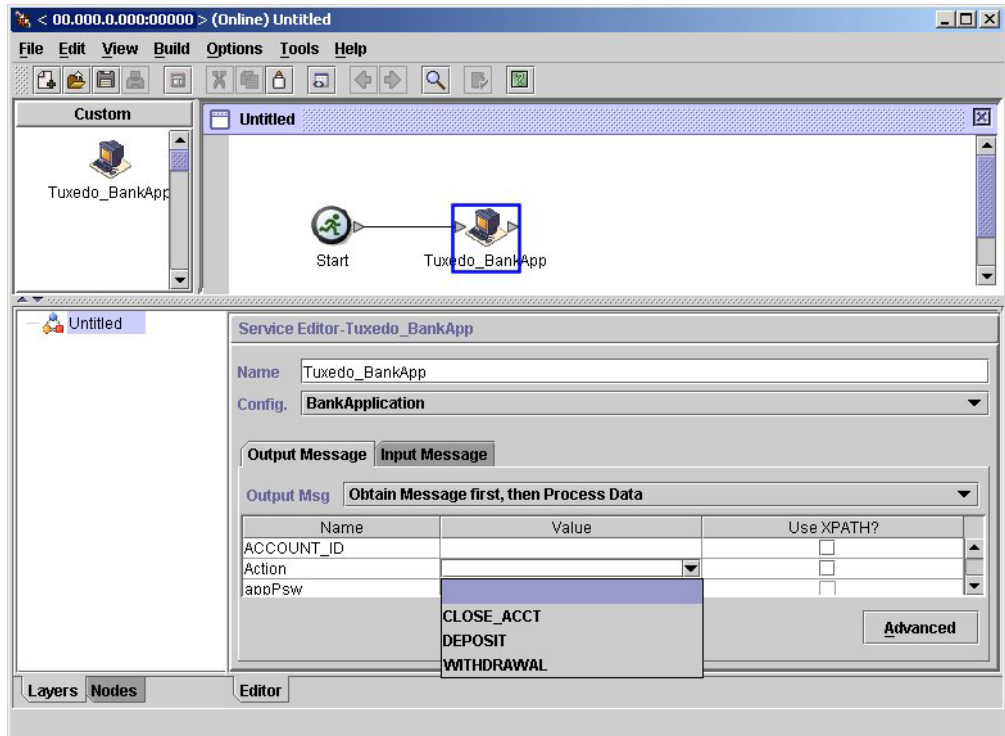
# Chapter 14. Begin Transaction Service

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

| System name | Begin Transaction Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services |
| Description | Starts a transaction. |
| Business usage | Use this service to mark the start of a multi-step transaction in a business process. All of the steps after this service are part of one transaction and are committed or rolled back at the same time. This service should be used with the End Transaction service. The Begin Transaction service can also be used for global transactions where the transaction spans across adapters such as the MQ and Lightweight JDBC adapters. |
| Usage example | Consider a customer ordering items online and using a shopping cart. The merchant's business process:<br><br>1. Gets a document containing shopping cart information (customer information, product details).<br>2. Inserts the customer information to the customer table.<br>3. Inserts product details to the invoice table (product_id, quantity purchased).<br>4. Updates the quantity of the product from the inventory table as the products are sold.<br><br>If step 4 fails (update the quantity), but steps 1 and step 2 were committed. The merchant's inventory table is now inaccurate, because the items were reported as sold, but the items were not deducted from the inventory table. The Begin Transaction service and End Transaction service make it possible to have a transaction where all the steps are committed successfully or all the steps are rolled back. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The Begin Transaction service should be used with the End Transaction service. |
| Application requirements | Nothing external is required to use this service. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | No |
| Returned status values | Returned status values:<br>• 0 – Success<br>• 1 – Error |
| Restrictions | N/A |
| Testing considerations | N/A |

## How the Begin Transaction Service Works

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

## Business Process Example

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

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

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

```
<process name="TestTran">
  <sequence>
    <operation name="Begin">
      <participant name="BeginTransactionService"/>
      <output message="Xout">
        <assign to="START_TRANSACTION">TRUE</assign>
        <assign to="DISTRIBUTED">TRUE</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="root_element">root</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="LightweightJDBCAdapterType">
      <participant name="LightweightJDBCAdapterQuery"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
       <assign to="schedHour">-1</assign>
       <assign to="result_name">result</assign>
       <assign to="schedDay">-2</assign>
       <assign to="sql">INSERT INTO CUSTOMER_LW (CUSTOMER_ID,
             SIR_NAME, FIRST_NAME, MIDDLE_INITIAL, LAST_NAME, ADDRESS, CITY,
             STATE, ZIP, PHONE_NUM, EMAIL, ORDERS_PLACED, EMPLOYER,
             WORK_PHONE, WORK_EMAIL WORK_ADDRESS1, WORK_ADDRESS2, WORK_CITY,
             WORK_STATE, WORK_ZIP) VALUES
             (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
             </assign>
       <assign to="param1" from="Testplan/Test4/Customer_ID/text()"></assign>
       <assign to="paramtype1">Integer</assign>
       <assign to="param2" from="Testplan/Test4/SirName/text()"></assign>
```

```
      <assign to="paramtype2">String</assign>
      <assign to="param3" from="Testplan/Test4/FirstName/text()"></assign>
      <assign to="paramtype3">String</assign>
      <assign to="param4" from="Testplan/Test4/MiddleInitial/text()"></assign>
      <assign to="paramtype4">String</assign>
      <assign to="param5" from="Testplan/Test4/LastName/text()"></assign>
      <assign to="paramtype5">String</assign>
      <assign to="param6" from="Testplan/Test4/Address/text()"></assign>
      <assign to="paramtype6">String</assign>
      <assign to="param7" from="Testplan/Test4/City/text()"></assign>
      <assign to="paramtype7">String</assign>
      <assign to="param8" from="Testplan/Test4/State/text()"></assign>
      <assign to="paramtype8">String</assign>
      <assign to="param9" from="Testplan/Test4/Zip/text()"></assign>
      <assign to="paramtype9">String</assign>
      <assign to="param10" from="Testplan/Test4/Phone_Num/text()"></assign>
      <assign to="paramtype10">String</assign>
      <assign to="param11" from="Testplan/Test4/Email/text()"></assign>
      <assign to="paramtype11">String</assign>
      <assign to="param12" from="Testplan/Test4/Orders_Placed/text()"></assign>
      <assign to="paramtype12">Integer</assign>
      <assign to="param13" from="Testplan/Test4/Employer/text()"></assign>
      <assign to="paramtype13">String</assign>
      <assign to="param14" from="Testplan/Test4/Work_Phone/text()"></assign>
      <assign to="paramtype14">String</assign>
      <assign to="param15" from="Testplan/Test4/Work_Email/text()"></assign>
      <assign to="paramtype15">String</assign>
      <assign to="param16" from="Testplan/Test4/Work_Address1/text()"></assign>
      <assign to="paramtype16">String</assign>
      <assign to="param17" from="Testplan/Test4/Work_Address2/text()"></assign>
      <assign to="paramtype17">String</assign>
      <assign to="param18" from="Testplan/Test4/Work_City/text()"></assign>
      <assign to="paramtype18">String</assign>
      <assign to="param19" from="Testplan/Test4/Work_State/text()"></assign>
      <assign to="paramtype19">String</assign>
      <assign to="param20" from="Testplan/Test4/Work_Zipcode/text()"></assign>
      <assign to="paramtype20">String</assign>
      <assign to="pool">mysqlPool</assign>
      <assign to="row_name">row</assign>
      <assign to="schedMinute">-1</assign>
      <assign to="query_type">ACTION</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
  <operation name="LightweightJDBCAdapterType">
    <participant name="LightweightJDBCAdapterQuery"/>
    <output message="LightweightJDBCAdapterTypeInputMessage">
      <assign to="schedHour">-1</assign>
      <assign to="result_name">result</assign>
      <assign to="schedDay">-2</assign>
      <assign to="sql">INSERT INTO INVOICE CUSTOMER_ID,PRODUCT_ID,QTY)
            VALUES (?, ?, ?) </assign>
      <assign to="param1" from="Testplan/Test4/Customer_ID/text()"></assign>
      <assign to="paramtype1">Integer</assign>
      <assign to="param2" from="Testplan/Test4/PRODUCT_ID/text()"></assign>
      <assign to="paramtype2">String</assign>
      <assign to="param3" from="Testplan/Test4/QUANTITY/text()"></assign>
      <assign to="paramtype3">Integer</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
  <operation name="LightweightJDBCAdapterType">
```

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

## Implementing the Begin Transaction Service

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

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

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

   ```
   myPool.transactional=true
   ```

## Output from Business Process to Service

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

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

# Chapter 15. BP Exception Service

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

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

## How the BP Exception Service Works

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

## Business Process Example

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

```
<operation name="generateException">
    <participant name="BPExceptionService"/>
    <output message="Xout">
      <assign to="exceptionCode">foo:timeout</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input> </operation>
```

## Implementing the BP Exception Service

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

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

## Configuring the BP Exception Service

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

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

# Chapter 16. BP Fault Log Adapter

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

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

## How the BP Fault Log Adapter Works

The BP Fault Log adapter logs business process error messages and assigns log levels to various types of messages. Specifying a log level helps categorize and prioritize the generated logs. You may find it useful to configure the BP Fault Log adapter to fail a business process if a service introduces invalid data, and to log a message indicating the source of the bad data. This feature can save you time and

help you identify problems within your business process so you can correct them. Specify a location on disk to send and view generated business process error message logs.

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

### Implementing the BP Fault Log Adapter

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

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

### Configuring the BP Fault Log Adapter

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

### Sterling Integrator Configuration

The following table describes the fields used to configure the BP Fault Log 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 three options available for service groups:<br>• None – this service will not be part of a service group.<br>• Create New Group – to create a new service group for this service type, enter a unique name.<br>• Select Group – select an existing service group for this service type from the list. |
| Logger Name(loggerName) | Name of the log. Required. The file created in the Logger Output Path is given this name and an extension of .log. |
| Logger Output Path(loggerPath) | Path used to write the log to a file. Required. Valid value is a valid full path. |
| Include Document Body(includeBody) | Whether to include the body of the data with the error message. Required. Valid values are:<br>• True – Include the error message and the body of the data.<br>• False – Include the error message only. |
| Logging Type | Select type of log. Required. Valid values are Normal and Rolling. |

## GPM Configuration

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

| Field | Description |
|---|---|
| Config | Name of the adapter configuration. |
| fail-on-error | Whether to cause the business process to fail if an error is logged. Optional. Valid values are Y and N. If the parameter is set to Yes, and the log-level (see next field description) is set to ERROR, the business process status is set to Error and the business process advanced status is set to the value in the log-message. This enables you to force a business process to fail (based on invalid business process data), when it may have ordinarily succeeded. |
| log-level | Type of log. Required. Values are:<br>• DEBUG – An error message for internal use<br>• INFO – Who the error message was sent to<br>• WARN – A significant error in the business process<br>• ERROR – A general error in the business process<br><br>**Note:** You can add custom log levels to this list in the GPM, or by editing a business process in the business process editor. |
| log-message | Message that you want to send to the log. Required. |
| maxLogSize | Maximum log size for rolling in bytes. Required if you selected Rolling as the log type in the service configuration. Valid value is a number greater than zero. |

# Chapter 17. BP Fault XML Log adapter

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

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

## Implementing the BP Fault XML Log adapter

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

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

# Configuring the BP Fault XML Log adapter

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| action | Valid values are:<br>• READ–Reads the specified log from the specified directory, and uses the loggerName value as a prefix of the log file to read and loggerPath value as the directory path of where the log files are located.<br>• WRITE–performs a normal log write operation. |
| combineLog | Valid values are:<br>• True – Logs will be combined. When more than one log file name matches the prefix specified by the loggerName value, these log files will be combined into well-formed XML and placed onto process data as a single document. The process data document name is the specified loggerName value.<br>• False – Logs are not combined. When more than one log file name matches the prefix specified by the loggerName value, each log file will be placed onto process data. The process data document names are the same as the original log file names. |
| fail-on-error | Whether to cause the business process to fail if an error is logged. Optional. Valid values are True and False. Used for WRITE operations only. If this parameter is set to True, and the log-level parameter is set to ERROR, the business process status is set to Error and the business process advanced status is set to the value in the log-message. This enables you to force a business process to fail (based on invalid business process data), when it may have ordinarily succeeded. |
| includeBody | Whether to include the body of the data with the error message. Required. Used for WRITE operations only. When the body is included, only the first 2048 bytes are included. Valid values are:<br>• True – Include the error message and the body of the data.<br>• False – Include the error message only. |
| log-level | Type of log. Required. Used for WRITE operations only. Values are:<br>• DEBUG – An error message for internal use<br>• ERROR – A general error in the business process<br>• INFO – Who the error message was sent to<br>• WARN – A significant error in the business process<br>**Note:** You can add custom log levels to this list in the GPM, or by editing a business process in the business process editor. |
| log-message | Message that you want to send to the log. Required. Used for WRITE operations only. |

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

## Business Process Example

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

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

# Chapter 18. BP Metadata Info Service

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

| System name | Business Process Metadata |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | Use this service to retrieve information about a particular business process such as: <br><br>• Business process definition ID <br>• Process name <br>• Description <br>• State <br>• Status <br>• Type <br>• Priority <br>• Persistence level <br>• Lifespan <br>• Business process definition version <br>• Storage type <br>• Recovery level <br>• Doc tracking flag <br>• Deadline <br>• Event level <br>• Sub/parent business process information <br>• Correlations <br><br>**Note:** This service can also be used to retrieve information about the business process that runs this service. |
| Business usage | Enables you to retrieve information about a business process. This information can then be made available to the system to be used in other processes. |
| Usage Example | See *Business Process Example*. |
| Preconfigured? | Yes. The preconfigured copy is called BPMetaDataInfoService. |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | None |
| Application requirements | None |
| Initiates business processes? | No |
| Invocation | Does not invoke a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values: <br><br>• Success <br>• Error |
| Restrictions | None |

| Persistence level | None |
|---|---|
| Testing considerations | None |

### Implementing the BP Metadata Info Service

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

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

### Configuring the BP Metadata Info Service

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

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

### Output from Service to Business Process

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

| Parameter | Description |
|---|---|
| WORKFLOW_ID | The current Workflow ID. Required. Specify a known instance, or obtain from process data such as if using this service in an on-fault. Default is current workflow ID. |
| MESSAGE_FROM_SERVICE | The user who send the message from service. Optional. Default is current user. |
| WFD_ID | The current WorkFlow Definition ID. Required. |
| WFD_VERSION | Current BP version. Required. |
| WFD_NAME | Current BP name. Required. |
| WFD_DESCRIPTION | Current BP description. Required. |
| WFD_STATE | Current BP state; only show if the disposition is set. Optional. |

| Parameter | Description |
|---|---|
| WFD_STATUS | Current BP status; only show if the disposition is set. Optional. |
| WFD_TYPE | Current BP type. Required. |
| WFD_PRIORITY | Current BP priority. Required. |
| WFD_PERSISTENCE_LEVEL | Current BP persistence level. Required. |
| WFD_LIFE_SPAN | Current BP lifespan. Required. |
| WFD_STORAGE_TYPE | Current Storage type. Required. |
| WFD_RECOVERY_LEVEL | Current Recovery level. Required. |
| WFD_DOC_TRACKING_FLAG | Current Document tracking flag. Required. |
| WFD_DEADLINE_INTERVAL | Current Deadline interval. Optional. |
| WFD_EVENT_LEVEL | Current Event Level. Optional. |

## Output from Business Process to Service

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

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

## Business Process Example

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

```
<process name="BPMetaDataInfoService_01">
  <sequence name="simple">
   <operation name="1">
      <participant name="BPMetaDataInfoService"/>
      <output message="Xout">
      <assign to="WORKFLOW_ID">9</assign>
      <assign to="DISPOSITION">true</assign>
      <assign to="LINKAGE">true</assign>
      <assign to="CORRELATION">true</assign>
      <assign to="TRACE">true</assign>
      <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
      <assign to="." from="*"></assign>
      </input>
    </operation>
   </sequence>
</process>
```

# Chapter 19. BP Report Service

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

| System name | BP Report Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services |
| Description | Use the BP Report service to generate reports that contain a list of business process instance IDs that are not longer running and their states match the state you specify in the business process configuration. You can specify the following states of the business processes in the business process configuration:<br>• Halting<br>• Active<br>• Waiting_on_io |
| Business usage | Use this service to generate a list of business processes in one or more of the following states:<br>• Halting<br>• Active<br>• Waiting_on_io |
| 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 invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• wfc.setBasicStatus(WorkFlowContext.SUCCESS)<br>• wfc.setBasicStatus(WorkFlowContext.ERROR) |
| Restrictions | N/A |
| Persistence level | System Default |
| Testing considerations | None |

## Output from Business Process to Service

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

| Field | Description |
| --- | --- |
| STATE | Specify the state of the business process for which you want to generate a report. Requiired. Valid values are:<br><br>• Halting<br>• Active<br>• Waiting_on_io<br><br>The default value is Active. |
| showReport | View the details in the status report, such as, the queue the business process was or is in. |
| termInvalidWFD | Terminate a business process in one of the following states when the business process definition no longer exists in the system:<br><br>• halted<br>• interrupted_auto<br>• interrupted_man |

## Business Process Example

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

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

# Chapter 20. Cache Refresh Service

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

| System name | CacheRefreshServiceType |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Process Controls |
| Description | The Cache Refresh service allows a user to refresh cached customer defined property file information without restarting the Sterling Integrator. |
| Business usage | As a performance enhancement, you may write a business process that will pick up and cache information from a customer-defined properties file. When the same data is frequently referenced, caching it saves the time that would have been used for repeated lookups. The Cache Refresh service allows updated property file information in the cache to be refreshed without a restart of the system. |
| Usage example | For business reasons, information, such as a fax number, needs to be included in output documents. This information can be stored in the properties file and cached. When this information needs to be changed, such as a new fax number in this example, after the properties file has been changed, the cache must be refreshed or the system restarted. This service can be used to refresh the cache without a system restart. **Note:** Refreshing the cache makes updated property file information instantly available to all instances of business processes that reference it. |
| Preconfigured? | Yes. A CacheRefreshService configuration is available. However, values for required fields need to be entered when the Cache Refresh service is used in a business process. |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | None |
| Application requirements | None |
| Initiates business processes? | No. |
| Invocation | The Cache Refresh service is invoked by a business process. |
| Business process Considerations | The properties file needs to be referenced in the business process using an XPath function (sci-get-property). To run this service on a schedule, create a business process containing this service and then schedule the business process. |
| Returned status values | None |

| Restrictions | The referenced customer property file must either exist in the <Sterling Integrator>/properties directory, or the <Sterling Integrator>/properties/servers.properties file must contain a tag/value pair with "FileTag=<the full file description>". Only customer property files can be refreshed using this service. Cached Sterling Integrator property file information should not be updated while Sterling Integrator is running. |
|---|---|
| Persistence level | System Default |
| Testing considerations | None |

## Cache Refresh Service Operation

When invoked, the Cache Refresh service:

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

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

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

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

## Implementing the Cache Refresh Service

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

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

## Configuring the Cache Refresh Service

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

| Field | Description |
|---|---|
| Config | Unique and meaningful name for the service configuration. Required. |
| cache_name | Name of the property file or the property file tag. Required. Valid value: any string.<br>**Note:** This name must match the PropertyFileTag in sci-get-property. For more information, see *Referencing the Properties File*. |
| cache_type | Type of cache to use. Required. Valid value: Properties |

## Output from Business Process to Service

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

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

## Business Process Example

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

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

## Referencing the Properties File

XPath Function Syntax:

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

Where:

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

## Example:

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

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

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

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

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

# Chapter 21. Connect:Direct Requester Adapter

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

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

## Implementing the Connect:Direct Requester Adapter

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

1. Create a Connect:Direct Requester adapter configuration. For information, see *Managing Services and Adapters*.

2. Configure the Connect:Direct Requester adapter. For information, see *Configuring the Connect:Direct Requester Adapter*.
3. Use the Connect:Direct Requester adapter in a business process.

## Configuring the Connect:Direct Requester Adapter

To configure the Connect:Direct Requester adapter, you must specify settings for the following fields in 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 | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are: <br>• None – You do not want to include this configuration in a group at this time (default)<br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br>• 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.<br><br>For more information about service groups see *Managing Services and Adapters*. |
| Remote Connect:Direct Server Address | The IP address of the remote Connect:Direct Server to connect to. Required. Valid values are 0.0.0.0 to 255.255.255.254 (default is 127.0.0.1). |
| Remote Connect:Direct Server Port | The port number that the remote Connect:Direct Server is listening on for API connections. Required. Valid are any numeric values (default is 1364). |
| Remote Connect:Direct User ID | The user ID used to log on to the Connect:Direct server. Required. |
| Remote Connect:Direct User Password | The password used to log on to the Connect:Direct server. Required. |
| Select Connect:Direct Perimeter Services Option | The Perimeter server to be used. Required. Default value is Local Node1. |
| Encryption using Secure+ | Whether Connect:Direct Secure+ should be active. Required. Valid values are Enabled and Disabled (default). If Enabled is selected for the Encryption using Secure+ parameter, then the following fields are displayed:<br>• CA Certificates<br>• Cipher Suites<br>• SSL or TLS |
| CA Certificates | List of Secure+ CA certificates. Required if Select Connect: Direct Secure+ Option is enabled. |

| Field | Description |
| --- | --- |
| Cipher Suites | List of Secure+ Cipher suites. Required if Select Connect: Direct Secure+ Option is enabled. |
| SSL or TLS | Whether Secure Sockets Layer (SSL) or TLS should be used. Required if Select Connect:Direct Secure+ Option is enabled. Valid values are SSL (default) and TLS. |

# Chapter 22. Connect:Direct Requester Select Process Service

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

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

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

## Implementing the Connect:Direct Requester Select Process Service

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

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

## Configuring the Connect:Direct Requester Select Process Service

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

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

## Parameters Passed from Business Process to Service

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

| Parameter | Description |
|---|---|
| CDRequesterName | Specifies the Connect:Direct Requester adapter to use with this service. Required. |
| ProcessName | Specifies the 1–8 character name of the process. Optional. Valid value is alphanumeric that begins with an alpha character. |
| ProcessNumber | Specifies the system-assigned number of the process. Optional. Valid value range is 1–99999. |
| Queue | Specifies which queue to select processes from. Optional. Valid values are:<br><br>• Hold<br><br>• Wait<br><br>• Timer<br><br>• Exec |
| Snode | Specifies the secondary node to be used in this process. The secondary node name is a 1–16 alphanumeric character name that is defined in the network map. Optional. Valid values are alphanumeric characters or nationals (@#$.). |

| Parameter | Description |
|-----------|-------------|
| Status | Selects processes for viewing according to status. Optional. Valid values are:<br><br>• Execution - Process is running.<br>• Pending Execution - Process is selected to run and startup is in progress.<br>• Waiting Connection - Process is ready to run, but all available connections to the SNODE are in use.<br>• Waiting Start Time - Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired.<br>• Held Suspension - Operator issued a delete process request with Hold set to Yes.<br>• Timer Retry - Process error occurred and the process was moved to the Timer queue in Timer Retry status with short-term and long-term wait times beginning.<br>• Held for Call - Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in Waiting Connection status. The process is placed in the Execution queue when it is selected to run.<br>• Held Due to Error - session error or other abnormal condition occurred.<br>• Held Initially - Process was submitted with the Hold option set to Yes.<br>• Held By Operator - Change process request with Hold set to Yes was issued.<br>• Held By Retain - Process was submitted with retain after execution set to Yes or Initial. |
| Submitter | Specifies the node name and user ID of the user that submitted the process. Separate the node name and user ID with a comma; for example, (atlanta, user1), (atlanta, user2), (atlanta, user3). Optional. |
| UserId | User ID used to connect to the Connect:Direct server. Overrides the user ID specified in the Connect:Direct Requester adapter. Optional. |
| Password | Password used to connect to the Connect:Direct server. Overrides the password specified in the Connect:Direct Requester adapter. Optional. |

## Parameters Passed from Service to Business Process

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

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

| Parameter | Description |
|---|---|
| ConditionCode | Specifies the return code values associated with step termination. Optional. Valid codes are:<br>• 0 – Successful execution of the process.<br>• 4 – A warning error was encountered. The statement probably finished normally, but you should verify the execution results.<br>• 8 – An error occurred during process execution.<br>• 16 – A severe error occurred during process execution. |
| DestinationDisposition1 | Specifies what to do with the destination file after a copy is complete. Optional. The destination DISP valid values are:<br>• NEW – Creates a new file on the destination node.<br>• RPL – Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node.<br>• MOD – Appends data to the end of an existing file for which you have exclusive rights. |
| DestinationDisposition2 | Specifies the disposition of the destination file after a successful process step termination that results in a zero completion code. Optional. The destination DISP valid values are:<br>• NEW – Creates a new file on the destination node.<br>• RPL – Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node.<br>• MOD – Appends data to the end of an existing file for which you have exclusive rights. |
| DestinationDisposition3 | Specifies the disposition of the destination file after an abnormal process step termination that results in a non-zero completion code. Optional. The destination DISP valid values are:<br>• NEW–Creates a new file on the destination node.<br>• RPL–Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node.<br>• MOD –Appends data to the end of an existing file for which you have exclusive rights. |
| DestinationFile | Specifies the destination file used in the process. Optional. |
| EncryptionAlgorithmName | Specifies the name of the encryption algorithm. Optional. |
| ExecutionPriority | Priority under which the operating system thread that executes Connect:Direct runs. Optional. Valid values are 1-15.<br>**Note:** Applies to Windows only. |
| ExtendedCompression | Extended compression option. Optional. |
| Feedback | Feedback code for the module. Optional. Valid value depends on the module that creates it. |
| FromNode | Node that sent the file. Optional. Valid values are S–SNODE and P–PNODE. |
| Function | Function being performed. Optional. |

| Parameter | Description |
|---|---|
| Hold | Holds status of a process. Optional. Valid values are:<br>• No – Do not place process in the Hold queue. It is runs as soon as resources are available.<br>• Yes – Hold process in queue in Held Initially (HI) status until it is explicitly released.<br>• Call – Hold process until the SNODE, as specified in the process SNODE parameter, connects to the PNODE. The process is then released for execution. The process is also released when another process on the PNODE connects to the SNODE. |
| LocalNode | Node that processed the file. Optional. Valid values are S–SNODE and P–PNODE. |
| LogDateTime | Specifies the date and time that the statistics record was written to the statistics log. Optional. |
| MessageData | Specifies the short text associated with the message ID. Optional. |
| MessageId | Current message ID for the process. Optional. |
| Pnode | Primary node. Optional. |
| PnodePlexClass | Pnode class that directs the process to only certain servers in a Connect:Direct/Plex environment. Optional. |
| Priority | Priority assigned to the process. The lower the number the higher the priority. Optional. Valid values are 1-15. |
| ProcessName | Process name. Optional. |
| ProcessNumber | Process number. Optional. |
| Queue | Queue containing the process. Optional. Valid queues are:<br>• All–All processes in the TCQ.<br>• Execution–Processes currently being executed.<br>• Hold–Processes that are either held by the user or operator or held due to execution errors.<br>• Timer–Processes that are scheduled to be executed later, or processes in time retry due to session errors.<br>• Wait–Processes that are eligible for execution and are awaiting selection. |
| RecordCount | Number of records in the file that was sent. Optional. |
| Restart | Whether Restart is activated for this process. Optional. |
| Retain | Whether Connect:Direct retains a copy of a process after it is executed. Optional. Valid values are:<br>• Initial – Specifies to retain the process in the Hold queue for execution every time that Connect:Direct initializes.<br>• No – Specifies not to retain the process after it is executed.<br>• Yes – Specifies to retain the process in the Hold queue after it is executed. You can release the process for execution later or delete it. |
| ScheduledDateTime | Date and time that a process is scheduled to be executed. Optional. |
| SecureEnabled | Whether Secure+ is activated for the process. Optional. |

| Parameter | Description |
|---|---|
| Signature | Whether digital signature is activated. Optional. |
| Snode | Secondary node used for the process. Optional. |
| SnodePlexClass | Snode class that directs the process to only certain servers in a Connect:Direct/Plex environment. Optional. |
| SourceDisposition1 | Access to the source file during a copy operation. Optional. Source DISP valid values are:<br><br>• SHR – The file can be opened for read-only access while it is being copied.<br>• OLD – The file cannot be opened during the transfer. |
| SourceDisposition2 | Disposition of the source file after a successful process step termination that results in a zero completion code. Optional. Source DISP valid values are:<br><br>• SHR – The file can be opened for read-only access while it is being copied.<br>• OLD – The file cannot be opened during the transfer. |
| SourceDisposition3 | Disposition of the source file after an abnormal process step termination that results in a non-zero completion code. Optional. Source DISP valid values are:<br><br>• SHR – The file can be opened for read-only access while it is being copied.<br>• OLD – The file cannot be opened during the transfer. |
| SourceFile | Source file used in the process. Optional. |
| StandardCompression | Standard compression option. Optional. |

| Parameter | Description |
|---|---|
| Status | Process status. Optional. Valid statuses are:<br><br>• EX (Execution) Process is executing.<br>• PE (Pending Execution) Process is selected for execution and startup is in progress.<br>• WC (Waiting Connection) Process is ready to execute, but all available connections to the SNODE are in use.<br>• WA (Waiting Start Time) Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired.<br>• HS (Held Suspension) Operator issued a delete process request with Hold set to Yes.<br>• RE (Timer Retry) Process error occurred and the process was moved to the Timer queue in RE status with short-term and long-term wait times beginning.<br>• HC (Held for Call) Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in WC status. The process is placed in the Execution queue when it is selected for execution.<br>• HE (Held Due to Error) session error or other abnormal condition occurred.<br>• HI (Held Initially) Process was submitted with the Hold option set to Yes.<br>• HO (Held By Operator) Change process request with Hold set to Yes was issued.<br>• HR (Held By Retain) Process was submitted with retain after execution set to Yes or Initial.<br>• RS (Restart) Process is restarting. |
| StepName | Current process step. Optional. |
| SubmitNode | Node from which the process was submitted. Optional. |
| Submitter | User ID that submitted the process. Optional. |
| SubmitterNode | Node from which the process was submitted. Optional. |
| XmitBytes | Number of bytes transmitted. Optional. |
| XmitRUs | Number of network buffers transmitted. Optional. |

## Business Process Example

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

```
<process name="CDInterop_testSelectProcess_GENERIC">
  <sequence>
    <operation name="CDSelectProcess">
      <participant name="CDRequesterSelectProcess"/>
      <output message="CDSelectProcess">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="SelectProcResults" from="*"></assign>
      </input>
    </operation>
<!-- Move DOM into PrimaryDocument to setup the File System Adapter -->
    <assign to="." from="DOMToDoc(/ProcessData, 'PrimaryDocument', 'no',
```

```
                'ProcessData')"></assign>
<!-- Load the process id used for unique naming of the process data output
        file-->
    <operation name="GetProcessId">
      <participant name="This"/>
      <output message="thisRequest">
      </output>
      <input message="thisResponse">
          <assign to="ConfigData/wfId" from="//INVOKE_ID_LIST/text()"
              append="true"/>
      </input>
    </operation>
<!-- FSA to Extract ProcessData -->
    <operation name="File System Adapter">
      <participant name="CDInteropTestFSA"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_EXTRACT</assign>
        <assign to="appendOnExtract">false</assign>
        <assign to="assignedFilename" from="concat(ConfigData/wfId/text(),
            '_ProcessDataOut.xml')"/>
        <assign to="assignFilename">true</assign>
        <assign to="bootstrap">false</assign>
        <assign to="extractionFolder">/usr/home/out</assign>
        <assign to="useSubFolders">false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <onFault>
    <sequence name="OnFault">
<!-- Move DOM into PrimaryDocument to setup the File System Adapter -->
        <assign to="." from="DOMToDoc(/ProcessData, 'PrimaryDocument','no',
              'ProcessData')"></assign>
<!-- Load the process id used for unique naming of the process data output
      file -->
        <operation name="GetProcessId">
          <participant name="This"/>
          <output message="thisRequest">
          </output>
          <input message="thisResponse">
    <assign to="ConfigData/wfId" from="//INVOKE_ID_LIST/text()"
append="true"/>
          </input>
        </operation>
<!-- FSA to Extract ProcessData -->
        <operation name="File System Adapter">
          <participant name="CDInteropTestFSA"/>
          <output message="FileSystemInputMessage">
            <assign to="Action">FS_EXTRACT</assign>
            <assign to="appendOnExtract">false</assign>
            <assign to="assignedFilename"
from="concat(ConfigData/wfId/text(),
                      '_ProcessDataOut.xml')"/>
            <assign to="assignFilename">true</assign>
            <assign to="bootstrap">false</assign>
            <assign to="extractionFolder">/usr/home/out</assign>
            <assign to="useSubFolders">false</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>
```

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

# Chapter 23. Connect:Direct Requester Select Statistics Service

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

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

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

### Implementing the Connect:Direct Requester Select Statistics Service

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

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

### Configuring the Connect:Direct Requester Select Statistics Service

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

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

## Parameters Passed from Business Process to Service

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

| Parameter | Description |
|---|---|
| CDRequesterName | Specifies the Connect:Direct Requester adapter to use with this service. Required. |
| ConditionCodeOperation | Comparison operator used to select statistics based on the return code value associated with step termination. Optional. Valid values are: <br>• Equal <br>• Greater Than <br>• Greater Than or Equal To <br>• Less Than <br>• Less Than or Equal To <br>• Not Equal To |
| ConditionCodeValue | Value of the return code associated with step termination. Optional. Valid values are: <br>• 0 – Successful execution of the process. <br>• 4 – A warning error was encountered. The statement probably finished normally, but you should verify the execution results. <br>• 8 – An error occurred during process execution. <br>• 16 – A severe error occurred during process execution. |
| DestinationFile | Destination file used in the process. Optional. |
| ProcessNumber | System-assigned number of the process. Optional. Valid value range is 1–99999. |
| ProcessName | The 1–8 character name of the process. Optional. Valid value is alphanumeric that begins with an alpha character. |

| Parameter | Description |
|---|---|
| RecordCategory | Which record category to select statistics from. Optional. Valid values are:<br>• CAPR – Process records<br>• CAEV – Event records |
| RecordId | Which record ID to select statistics from. Optional. |
| Snode | Secondary node to be used in this process. The secondary node name is a 1–16 alphanumeric character name that is defined in the network map. Optional. |
| SourceFile | Source file used in the process. Optional. |
| StartDate | Day or date to start selecting statistics records. Optional. |
| StartTime | Time to start selecting statistics records. Format is HH:MM [XM]. Optional. |
| StopDate | Day or date to stop selecting statistics records. Optional. |
| StopTime | Time to stop selecting statistics records. Format is HH:MM [XM]. Optional. |
| Submitter | User ID that submitted the process. Optional. |
| UserId | User ID used to connect to the Connect:Direct server. Overrides the user ID specified in the Connect:Direct Requester adapter. Optional. |
| Password | Password used to connect to the Connect:Direct server. Overrides the password specified in the Connect:Direct Requester adapter. Optional. |

## Parameters Passed from Service to Business Process

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

| Parameter | Description |
|---|---|
| AliasMember | Alias name of the member copied. Optional. |
| BytesRead | Number of bytes read from the source file. Optional. |
| BytesReceived | Number of bytes received by the destination file. Optional. |
| BytesSent | Number of bytes sent to the destination file. Optional. |
| BytesWritten | Number of bytes written to the destination file. Optional. |
| Checkpoint | Whether checkpointing is activated for this process. Optional. |
| Class | Node-to-node session on which a process can be executed. Optional. |
| ConditionCode | Return code values for the step termination. Optional. Valid values are:<br>• 0 – Successful process execution.<br>• 4 – A warning error was encountered. The statement probably completed normally, but you should verify the execution results.<br>• 8 – An error occurred during process execution.<br>• 16 – A severe error occurred during process execution. |

| Parameter | Description |
|---|---|
| ControlBlockEncryptionAlgorithm | Algorithm used to encrypt Connect:Direct control blocks used for strong authentication. This is the first algorithm ID in the PNODE list that is also in the SNODE list. Optional. |
| CurrentSignatureVerified | Whether the current encryption key was used for verifying the digital signature. Optional. |
| DestinationDisposition1 | What to do with the destination file after a copy is complete. Optional. Valid values are:<br>• NEW – Creates a new file on the destination node.<br>• RPL – Creates a new file on the destination node or, if the file already exists, replaces the named file on the destination node.<br>• MOD – Appends data to the end of an existing file for which you have exclusive rights. |
| DestinationDisposition2 | Disposition of the destination file after a successful process step termination that results in a zero completion code. Optional. |
| DestinationDisposition3 | Disposition of the destination file after an abnormal process step termination that results in a non-zero completion code. Optional. |
| DestinationFile | Name of the destination file. Optional. |
| ExecutionPriority | Priority under which the operating system thread that executes Connect:Direct runs. Optional.<br>**Note:** Applies to Windows only. |
| ExtendedCompression | Extended compression option. Optional. |
| Feedback | Feedback code for the module. Applies to Windows only. Valid values depend on the module that creates it. |
| FromNode | Node that sent the file. Optional. Valid values are S–SNODE and P–PNODE. |
| Function | Function being performed. Optional. |
| Hold | Hold status of a process. Optional. Valid values are:<br>• No – The process is not placed in the Hold queue. It is executed as soon as resources are available.<br>• Yes – The process is held in the Hold queue in Held Initially (HI) status until it is explicitly released.<br>• Call – The process is held until the SNODE, as specified in the process SNODE parameter, connects to the PNODE. The process is then released for execution. The process is also released when another process on the PNODE connects to the SNODE. |
| LinkFail | Whether a link failure occurred during transmission. Optional. |
| LocalConditionCode | Condition code produced by the local node. Optional. |
| LocalMessageId | Message ID produced by the local node. Optional. |
| LocalNode | Node that processed the file. Optional. Valid values are S–SNODE and P–PNODE. |
| LogDateTime | Date and time that the statistics record was written to the statistics log. Optional. |
| MemberName | Name of the member copied. Optional. |

| Parameter | Description |
|---|---|
| MergeEncryptionAlgorithm | Merged data encryption algorithm resulting from the merger of the PNODE and SNODE encryption algorithms. Optional. |
| MergeSignature | Merged results from the digital signature settings for the PNODE and SNODE. If digital signatures are enabled for either the PNODE or the SNODE, then digital signatures are used for the session. If digital signatures are not enabled for both the PNODE and SNODE, digital signatures are not used. Optional. |
| MessageId | Current message for the process. Optional. |
| MessageShortText | Short text for the message ID. Optional. |
| OtherConditionCode | Condition code produced by the other (remote) node. Optional. |
| OtherMessageId | Message ID produced by the other (remote) node. Optional. |
| Pnode | Primary node. Optional. |
| PnodeAccounting | User-entered PNODE accounting information. Optional. |
| PnodeEncryptionAlgorithmList | Data encryption algorithm used on the PNODE. Optional. |
| PnodeEncryptionData | PNODE encryption data. Optional. |
| PnodePlexClass | PNODE Connect:Direct/Plex class. Optional. |
| PnodeSignature | Whether digital signatures are enabled for the PNODE. Optional. |
| PreviousSignatureVerified | Whether the previous encryption key was used for verifying the digital signature. Optional. |
| Priority | Priority assigned to the process. The lower the number the higher the priority. Optional. |
| ProcessName | Process name. Optional. |
| ProcessNumber | Process number. Optional. |
| Queue | Queue containing the process. Optional. |
| RecordCategory | Whether the record is related to an event or a process. Optional. Valid values are:<br>• CAEV – The record is related to a Connect:Direct event, such as a Connect:Direct shutdown.<br>• CAPR – The record is related to a Connect:Direct process. |
| RecordId | Record type indicator. Optional. |
| RecordsRead | Number of records read from the source file. Optional. |
| RecordsWritten | Number of records written to the destination file. Optional. |
| Restart | Whether the process was restarted. Optional. |
| Retain | Whether Connect:Direct retains a copy of a process after it is executed. Optional. |
| RUSize | Size of buffers received by the destination file. Optional. |
| RUsReceived | Number of buffers received by the destination file. Optional. |
| RUsSent | Number of buffers sent to the destination file. Optional. |

| Parameter | Description |
|---|---|
| ScheduledDateTime | Date and time that a process is scheduled to be executed. Optional. |
| SecurePlusEnabled | Indicates that Secure+ is activated for the process. Optional. |
| ServerName | Connect:Direct node name. Optional. |
| Snode | Name of the secondary node. Optional. |
| SnodeAccounting | User-entered SNODE accounting information. Optional. |
| SnodeEncryptionAlgorithmList | Data encryption algorithm used on the SNODE. Optional. |
| SnodeEncryptionData | SNODE encryption data. Optional. |
| SnodePlexClass | SNODE Connect:Direct/Plex class. Optional. |
| SnodeSignature | Whether digital signatures are enabled for the SNODE. Optional. |
| SourceDisposition1 | Access to the source file during a copy operation. Optional. |
| SourceDisposition2 | Disposition of the source file after following a successful process step termination that results in a zero completion code. Optional. |
| SourceDisposition3 | Disposition of the source file after following an abnormal process step termination that results in a non-zero completion code. Optional. |
| SourceFile | Name and location of the source file. Optional. |
| SourceMember | Source name of the member copied. Optional. |
| StandardCompression | Specifies the standard compression option. Optional. |
| StartDateTime | Date and time that the process started. Optional. |

| Parameter | Description |
|---|---|
| Status | Process status. Optional. Valid values are:<br><br>• Execution - Process is executing.<br>• Pending Execution - Process is selected for execution and startup is in progress.<br>• Waiting Connection - Process is ready to execute, but all available connections to the SNODE are in use.<br>• Waiting Start Time - Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired.<br>• Held Suspension - Operator issued a delete process request with Hold set to Yes.<br>• Timer Retry - Process error occurred and the process was moved to the Timer queue in Timer Retry status with short-term and long-term wait times beginning.<br>• Held for Call - Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in Waiting Connection status. The process is placed in the Execution queue when it is selected for execution.<br>• Held Due to Error - Session error or other abnormal condition occurred.<br>• Held Initially - Process was submitted with the Hold option set to Yes.<br>• Held By Operator - Change process request with Hold set to Yes was issued.<br>• Held By Retain - Process was submitted with retain after execution set to Yes or Initial.<br>• Restart - Process is restarting. |
| StepName | Process step. Optional. |
| StopDateTime | Date and time that the process stopped. Optional. |
| SubmitDateTime | Date and time that the process was submitted. Optional. |
| SubmitNode | Node from which the process was submitted. Optional. |
| Submitter | User ID that submitted the process. Optional. |
| SubmitterNode | Node from which the process was submitted. Optional. |
| Sysopts | Platform-specific system operations. These parameters specify the data type, translation tables, inherited rights, attributes, and trustees. Optional. |
| TargetMember | Target name of the member copied. Optional. |
| Translation | Whether the data was translated. Optional. |

## Business Process Example

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

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

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

# Chapter 24. Connect:Direct Requester Submit Process Service

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

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

## Implementing the Connect:Direct Requester Submit Process Service

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

1. Create a Connect:Direct Requester Submit Process service configuration. For information, see *Managing Services and Adapters*.

2. Configure the Connect:Direct Requester Submit Process service. For information, see *Configuring the Connect:Direct Requester Submit Process Service*.

3. Use the Connect:Direct Requester Submit Process service in a business process.

## Configuring the Connect:Direct Requester Submit Process Service

To configure the Connect:Direct Requester Submit Process service, you must specify settings for the following fields in 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 | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are: <br>• None – You do not want to include this configuration in a group at this time (default) <br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. <br>• 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. <br><br>For more information about service groups see *Managing Services and Adapters*. |
| CDRequesterName | Specifies the Connect:Direct Requester adapter to use with this service. Required. |

## Parameters Passed from Business Process to Service

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

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

| Parameter | Description |
|-----------|-------------|
| Hold | Specifies the Hold status of a process. Optional. Valid values are:<br><br>• No – Do not place process in the Hold queue. It is executed as soon as resources are available (default).<br><br>• Yes – Hold process in the Hold queue in Held Initially (HI) status until it is explicitly released.<br><br>• Call – Hold process until the SNODE, as specified in the process SNODE parameter, connects to the PNODE. The process is then released for execution. The process is also released when another process on the PNODE connects to the SNODE. |
| MaxDelay | Specifies the time to wait until a process has executed. Optional. Valid values are:<br><br>• Unlimited – Wait indefinitely or until the process is complete.<br><br>• Queued – Wait until the process is submitted and placed on the queue.<br><br>• hh:mm:ss – Wait for a specific length of time.<br><br>• 0 – The process must be placed on the Exec queue immediately or the submission of the process is considered unsuccessful. |
| NewName | Specifies the new name of the process. The default is the label on the process statement. Optional. |
| NotifyUser | Specifies the user ID to receive process completion messages. The user ID is notified through a Microsoft Exchange E-mail, a Windows dialog box, or a TSO notify. Optional. |
| PnodeAccounting | Specifies user data passed to the PNODE. Optional. |
| PnodePassword | Specifies the user password on the PNODE. This field is case-sensitive. Optional. |
| PnodeUserid | Specifies the user ID used as a security ID on the PNODE. This ID must be the name of an existing user account. This field is case-sensitive. Optional. |
| Priority | Specifies the priority of a process in the Transmission Control queue. Connect:Direct uses the Priority parameter for process selection. The lower the number, the higher the priority. A process with higher priority is selected for execution before a process with a lower priority. This parameter does not affect the priority during transmission. Optional. Valid values are 1–15 (default is 7). |
| ProcessDocumentId | Specifies the Document ID of the document that contains text of the process to be submitted to the Connect:Direct server. Optional (if both ProcessText and ProcessDocumentId are blank, then the primary document must contain the process text). |
| ProcessText | Specifies the text of the process to be submitted to the Connect:Direct server. Optional (If both ProcessText and ProcessDocumentId are blank, then the primary document must contain the process text). |

| Parameter | Description |
|-----------|-------------|
| Retain | Whether Connect:Direct retains a copy of a process after it is runs. If you specify Retain with a start time, the process is released to run at the specified time. Each time a retained process is released, Connect:Direct creates a copy with a new process number. The copy is executed, and the original process remains in the queue. Optional. Valid values are: <br><br> • Initial – Retain the process in the Hold queue for execution every time that Connect:Direct initializes. Do not specify a start time if you choose this option. <br><br> • No – Do not retain the process after it is executed (default). <br><br> • Yes – Retain the process in the Hold queue after it is executed. You can release the process for execution later or delete it. When you specify a start date and start time, set Retain to Yes to continually execute the process at the scheduled time. |
| Snode | Specifies the secondary node to be used in this process. The secondary node name is a 1–16 alphanumeric character name that is defined in the network map. Optional. Valid values are alphanumeric or nationals (@#$.). |
| SnodeAccounting | Specifies user data passed to the SNODE. Optional. |
| SnodePassword | Specifies the user password on the SNODE. This field is case-sensitive. Optional. |
| SnodeUserid | Specifies the user ID used as a security ID on the SNODE. This field is case-sensitive. Optional. |
| StartDate | Specifies the day or date to execute the process. Optional. |
| StartTime | Specifies the time to execute the process. Format is *HH:MM* [*XM*]. Optional. |
| Variables | Specifies the symbolic variables to be used in the process. Symbolic variables are text strings in a process that are replaced with predefined values when the process is executed. This allows you to easily change processes when these values change. For example, you can define a variable *&filename* and declare the value to be file1.txt. Whenever the process encounters the *&filename* variable, it substitutes file1.txt. If you want to use a different value, just change the *&filename* definition to the new file name, and the process will use that instead. Optional. |
| UserId | User ID used to connect to the Connect:Direct server. Overrides the user ID specified in the Connect:Direct Requester adapter. Optional. |
| Password | Password used to connect to the Connect:Direct server. Overrides the password specified in the Connect:Direct Requester adapter. Alphanumeric. Optional. |

## Parameters Passed from Service to Business Process

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

| Parameter | Description |
|---|---|
| Class | Determines the node-to-node session on which a process can be executed. Optional. Valid values are 1–10. |
| ConditionCode | Specifies the return code values associated with step termination. Optional. Valid values are:<br>• 0 – Successful execution of the process.<br>• 4 – A warning error was encountered. The statement probably finished normally, but you should verify the execution results.<br>• 8 – An error occurred during process execution.<br>• 16 – A severe error occurred during process execution. |
| ExecutionPriority | The priority under which the operating system thread that executes Connect:Direct runs. Valid values are 1–15. Optional.<br>**Note:** Applies to Windows only. |
| LogDateTime | Specifies the date and time that the statistics record was written to the statistics log. Optional. |
| MessageId | Specifies the current message ID for the process. Optional. |
| MessageText | Specifies the short text associated with the message ID. Optional. |
| Pnode | Specifies the primary node. Optional. |
| Priority | Specifies the priority assigned to the process. Optional. The lower the number the higher the priority. Valid values are 1–15. |
| ProcessName | Specifies the process name. Optional. |
| ProcessNumber | Specifies the process number. Optional. |
| Queue | Specifies the queue containing the process. Optional. Valid queue values are:<br>• All–All processes in the TCQ.<br>• Execution–Processes currently being executed.<br>• Hold–Processes that are either held by the user or operator or held due to execution errors.<br>• Timer–Processes that are scheduled to be executed later, or processes in time retry due to session errors.<br>• Wait–Processes that are eligible for execution and are awaiting selection. |
| Snode | Specifies the secondary node used for the process. |

| Parameter | Description |
|---|---|
| Status | Process status. Optional. Valid status values are:<br><br>• Execution -Process is executing.<br><br>• Pending Execution -Process is selected for execution and startup is in progress.<br><br>• Waiting Connection -Process is ready to execute, but all available connections to the SNODE are in use.<br><br>• Waiting Start Time -Process is waiting in the Timer queue because it was submitted with a start time or date that has not expired.<br><br>• Held Suspension - Operator issued a delete process request with Hold set to Yes.<br><br>• Timer Retry - Process error occurred and the process was moved to the Timer queue in Timer Retry status with short-term and long-term wait times beginning.<br><br>• Held for Call - Process was submitted with the Hold parameter set to Call. A session started from either node moves the process to the Wait queue in Waiting Connection status. The process is placed in the Execution queue when it is selected for execution.<br><br>• Held Due to Error - Session error or other abnormal condition occurred.<br><br>• Held Initially - Process was submitted with the Hold option set to Yes.<br><br>• Held By Operator - Change process request with Hold set to Yes was issued.<br><br>• Held By Retain - Process was submitted with Retain After Execution set to Yes or Initial. |
| Submitter | Specifies the user ID that submitted the process. Optional. |
| SubmitterNode | Specifies the node from which the process was submitted. Optional. |

## Business Process Example

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

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

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

# Chapter 25. Connect:Direct Server Begin Session Service

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

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

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

## Implementing the Connect:Direct Server Begin Session Service

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

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

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

## Configuring the Connect:Direct Server Begin Session Service

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

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

| Field | Description |
|---|---|
| Name | Name of the service configuration. Required. |
| Description | Description of service. Required. |

| Field | Description |
|-------|-------------|
| Select a Group | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |

## Parameters Passed from Business Process to Service

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

| Field | Description |
|-------|-------------|
| Number of Retry attempts for establishing a session(BeginSessionMaxRetries) | Number of times to retry creating a session. Optional. When specified here, the value overrides the values set in the Connect:Direct Server adapter configuration. Valid value is any number up to 9999. Default is 0. The following errors fail without any retries:<br><br>• Invalid required parameter<br>• Missing required parameter<br>• Adapter not found<br>• Remote node not in the netmap<br>• Connect:Direct unable to locate local server |
| Interval between session establishment attempts (minutes)(BeginSessionRetryInterval) | Time, in minutes, to wait between attempts to create a session. Optional. When specified here, the value overrides the values set in the Connect:Direct Server adapter configuration. Valid value is any number up to 99999999. Default is 1. |
| Number of short-term session retry attempts(ShortTermMaxRetries) | Specifies the maximum number of short-term session retry attempts that should be made when failures to connect occur. When specified here, the value overrides the values set in the Connect:Direct Server adapter configuration. Valid value: Any number up through 9999. Default is 0. Short-term retry activity is logged (if logging enabled). |
| Interval between short-term session retry attempts (seconds)(ShortTermRetryInterval) | Specifies the amount of time to wait between short-term session establishment attempts. When specified here, the value overrides the values set in the Connect:Direct Server adapter configuration. Valid value: Any number through 99999999. Short-term retry activity is logged (if logging enabled). |
| LocalCDNodeName | Specifies the name of the local Connect: Direct Server adapter. List includes all configurations of the Connect:Direct Server adapter. Required. |
| PlexClass | Specifies the PlexClass to be used on the remote Connect:Direct server node. Optional. |

| Field | Description |
|---|---|
| ProcessName | Specifies a user-designated name for the application business process that initiated the Connect:Direct Server Begin Session service. Optional. If not specified, the process name is generated dynamically. Maximum eight characters.<br>**Note:** If a ProcessName is specified, hyperlinks from the Connect:Direct Browser interface to the application are not active. |
| RemoteCDNodeName | Specifies the name of a Remote Connect:Direct Server node. This name must be same as the name of one of the nodes in the netmap of the adapter identified by LocalCDNodeName. Required. |
| RemotePasswd | Specifies the password to be used on Remote Connect:Direct Server Node. Optional. |
| RemoteUserId | Specifies the UserId to be used on remote Connect:Direct server node. Required except when:<br>1. Remote password not provided in the invoked BP.<br>2. Remote Connect:Direct Server is configured to accept an inbound connection from a designated PNodeName (Specified Connect:Direct server adapter name in the invoked BP) and PNodeId (the user executing the BP). |

## Parameters Passed from Service to Business Process

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

| Parameter | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| SessionBeginTime | Specifies the date and time when the session was established. |

## Business Process Example

The following business process example uses the Obscure Data -Process Data Values service to obscure the password in BPML. This example business process

will only succeed if an instance of the Obscure Data -Process Data Values service called "CDClientObscureParameter" has already been created with "cinterop" and "******* " (the password value) as values:

```
<process name="CDInterop_SessionBegin">
  <sequence name="CDServerSessionBeginGIS">
<operation name="Obscure Password">
          <participant name="CDClientObscureParameter"/>
          <output message="outmsg">
             <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
             <assign to="." from="*"></assign>
          </input>
       </operation>
    <!-- Begin Server Session -->
       <operation name="Connect:Direct Server Begin Session Service">
         <participant name="CDServerBeginSession"/>
         <output message="BeginSession">
           <assign to="." from="*"></assign>
           <assign to="LocalCDNodeName">TEST1_CDSERVER</assign>
           <assign to="RemoteCDNodeName">ZZ.AAA.ZZZZZ</assign>
           <assign to="RemoteUserId">auser</assign>
    <assign to="RemotePasswd" from="revealObscured(cinterop)"></assign>
         </output>
         <input message="inmsg">
           <assign to="BeginSessionResults" from="*"></assign>
         </input>
       </operation>
    <!-- End Server Session -->
       <operation name="CONNECT:Direct Server End Session Service">
         <participant name="CDServerEndSession"/>
         <output message="EndSession">
           <assign to="." from="*"></assign>
           <assign to="SessionToken"
    from="//BeginSessionResults/SessionToken/node()"></assign>
         </output>
         <input message="inmsg">
           <assign to="EndSessionResults" from="*"></assign>
         </input>
       </operation>
  </sequence>
</process>
```

# Chapter 26. Connect:Direct Server BP Response Service

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

| System Name | Connect:Direct Server BP Response Service |
| --- | --- |
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect:Direct |
| Description | This service is used to call back from a business process to a Connect:Direct Server adapter. |
| Business usage | This service is used as the last operation in a business process launched from a Connect:Direct Server adapter in response to a request from a remote Connect:Direct Server to copy-from-BP. The service performs a call-back on the Connect:Direct Server adapter that launched the business process to convey the primary document which is used as the response to send to the remote Connect:Direct server. (This is similar to the HTTP Respond service, which performs a callback on the HTTP Server adapter). This service uses session information that is pre-placed in the process data by the Connect:Direct Server adapter that launches the business process. Therefore the service can only be used in business processes launched by a Connect:Direct Server adapter. |
| Usage example | A remote Connect:Direct server initiates a session with a Connect:Direct Server adapter in Sterling Integrator and requests to copy (retrieve) the final primary document resulting from a business process launched by the Connect:Direct Server adapter. The name of the business process is provided in the request from the remote Connect:Direct node. The business process must be created in Sterling Integrator and have the BP Response Service as the last operation, so that the service can call back the adapter to respond to the request from the remote Connect:Direct node. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | Connect:Direct Server adapter. The Connect:Direct Server BP Response service can only be used in business processes launched by the Connect:Direct Server adapter. |
| Application requirements | Application requirements:<br>• Perimeter server<br>• Connect:Direct Windows 4.1.00 or later<br>• Connect:Direct UNIX 3.5.00 or later<br>• Connect:Direct OS/390 4.4.00 or later<br>• Connect:Direct OS/400 3.5.00 or later |
| Initiates business processes? | No |

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

## Implementing the Connect:Direct Server BP Response Service

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

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

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

## Configuring the Connect:Direct Server BP Response Service

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

## Parameters Passed from Business Process to Service

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

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

## Business Process Example

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

```
<process name="CDInterop_SnodeCallback_BP">
 <!-- A simple BP that may be kicked off at Snode during CDServer Copy
 from BP test (Sterling Integrator as Snode).
```

```
     !!NOTE!! :
This BP may ONLY be launched by the BusinessProcessInitiator class;
will fail if launched using execution manager.    -->
<sequence name="CDServerSnodeCallbackBP">
<!-- Read a file from disk using the file-system adapter which will become
          the primary document-->
<operation name="File System Adapter">
<participant name="CDInteropTestFSA"/>
<output message="FileSystemParams">
<assign to="Action">FS_COLLECT</assign>
<assign to="bootstrap">false</assign>
<assign to="collectionFolder">/install_dir/foo/bar/in</assign>
<assign to="useSubFolders">false</assign>
<assign to="filter" from="Samplefile.txt"/> <!-- file to read -->
<assign to="deleteAfterCollect">false</assign>
<assign to="." from="*"></assign>
</output>
<input message="FileSystemResults">
<assign to="." from="*"></assign>
</input>
</operation>
<!-- Perform callback on the adapter using session-token
from process-data -->
<!-- NOTE:This operation must be the last one in the BP -->
<operation name="CONNECT:Direct Server Begin Session Service">
<participant name="CDServerBPResponseServiceInst/>
<output message="BPResponseParams">
<assign to="." from="*"></assign>
</output>
<input message="inmsg">
<assign to="BPResponseResults" from="*"></assign>
</input>
</operation>
</sequence>
</process>
```

# Chapter 27. Connect:Direct Server CopyFrom Service

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

| System Name | Connect:Direct Server CopyFrom Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect:Direct |
| Description | This service copies a file from a remote Connect:Direct server. To ensure that the copy performed by this service can restart if a Sterling Integrator crash interrupts the copy, the business process (in which this service is used) must have the recovery level set to Auto-resume. If so marked, the business process and the copy are resumed automatically when Sterling Integrator restarts, if the remote Connect:Direct node is still active. When resumed, the copy is restarted from the beginning (or from the last checkpoint if configured). |
| Business usage | A business user would use this service to copy a file from a remote Connect:Direct. |
| Usage example | A trading partner copies a business document from a remote Connect: Direct outside the organization. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | Whenever the Connect:Direct Server CopyFrom service is used in a business process, the Connect:Direct Server Primitive CopyFrom service is exercised behind the scenes as part of a system business process to perform the actual copy operation. The Connect:Direct Server CopyFrom service has tight integration with the Connect:Direct Server Begin Session service. A session must be established between two Connect:Direct nodes using the Begin Session service to execute the CopyFrom service successfully. The Connect:Direct Server End Session service must be used to end the session established between the Connect:Direct nodes after the Connect:Direct Server CopyFrom service. |
| Application requirements | Application requirements:<br>• Perimeter server<br>• Connect:Direct Windows 4.1.00 or later<br>• Connect:Direct UNIX 3.5.00 or later<br>• Connect:Direct OS/390 4.4.00 or later<br>• Connect:Direct OS/400 3.5.00 or later |
| Initiates business processes? | No |
| Invocation | A user who has permission to perform this activity must start the business process that runs this service. |

| Business process context considerations | LocalStripBlanks and LocalXlate are optional parameters. Use these parameters only when you want to control non-standard behavior and you have a complete understanding of your environment, the local content, the target requirement, data translation, and character set conversion. If LocalStripBlanks and LocalXlate are not explicitly assigned in the business process (see example below), translation occurs automatically during transmission. Example of a BP that does not explicitly assign LocalStripBlanks and LocalXlate parameters: |
|---|---|

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

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

**Note:** If the LocalXlate parameter is set to 'Yes', the SysOpts parameter must also specify 'xlate (yes)'.Example of a BP that explicitly assigns LocalStripBlanks and LocalXlate parameters:

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

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

## Implementing the Connect:Direct Server CopyFrom Service

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

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

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

## Configuring the Connect:Direct Server CopyFrom Service

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |

| Field | Description |
|---|---|
| BinaryMode | Specifies the mode for Connect:Direct File transfer. Optional. Valid values are:<br><br>• Yes - simple binary mode<br>• No - (default) - text mode<br>• VB - a variable block binary transfer with Connect:Direct on an OS/390 |
| CheckpointInterval | Specifies the byte interval for checkpoint support, which allows restart of interrupted transmissions at the last valid checkpoint point and therefore reduces the time to retransmit the file. Optional. Valid value is number of bytes, in the format: *nnnnnnnnnn*. Sterling Integrator stores checkpoint information for 30 days, after which it is automatically purged. If your database is corrupted, purge this checkpoint information by performing a cold restart. To do this, edit the adapter configuration Server Start Option parameter to Cold Start. Once you have performed the cold start, edit the adapter configuration to change the setting back to Warm Start. You can also purge the checkpoint information from the database manually. |
| Compression | Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9. Increasing value indicates higher attempt at compression. Values greater than 2 may not result in further compression. |
| LocalDocumentName | Name of the document in Sterling Integrator to which the remote file should be copied. Optional. |

| Field | Description |
|---|---|
| LocalFromCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:<br><br>• Code page specified is supported by the local Java VM<br><br>• Parameter LocalToCodePage is defined<br><br>• Text mode file transfer (BinaryMode = No)<br><br>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:<br><br>`<assign to="LocalFromCodePage">`<br>`Cp1251`<br>`</assign>`<br>`<assign to="LocalToCodePage">`<br>`Cp1047`<br>`</assign>`<br><br>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalStripBlanks | Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default). |

| Field | Description |
| --- | --- |
| LocalToCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions: <br> • Code page specified is supported by the local Java VM <br> • Parameter LocalFromCodePage is defined <br> • Text mode file transfer (BinaryMode = No) <br><br> Optional. Valid value is the name of a codepage supported by the local Java VM. For example: <br><br> <assign to="LocalFromCodePage">Cp1251</assign> <br><br> <assign to="LocalToCodePage">Cp1047</assign>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalXlate | Specifies if the Connect:Direct Server adapter is to perform EBCDIC to ASCII translation before creating a new document from the received remote file. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No. |
| RemoteFileName | Name of the file that should be copied from the remote Connect:Direct server. Enter the legal characters in the file name. Required. |
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <br> • SessionId <br> • ProcessNumber <br> • ProcessName <br> • SubmitDateTime <br> • RemoteCDNodeName <br> • RemoteUserId <br> • RemotePasswd <br> • PlexClass <br> • MaxRestartRetries <br> • RestartRetryInterval |

| Field | Description |
|---|---|
| Sysopts | Remote system operations to be implemented on the remote Connect:Direct node. Optional. For example (for copying to Connect:Direct Windows only):<br><br>Datatype(text) xlate(yes)For more information, see your Connect:Direct documentation. |

## Parameters Passed from Business Process to Service

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

| Field | Description |
|---|---|
| BinaryMode | Specifies the mode for Connect Direct File transfer. Optional. Valid values are:<br>• Yes - simple binary mode<br>• No - (default) - text mode<br>• VB - a variable block binary transfer with Connect:Direct on an OS/390 |
| CheckpointInterval | Specifies the byte interval for checkpoint support, which enables restart of interrupted transmissions at the last valid checkpoint point and to reduce the time to retransmit the file. Optional. Valid value is number of bytes, in the format: *nnnnnnnnnn*. |
| Compression | Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9. |
| DCB | Specifies Data Control Block (DCB), a group of attributes specified as name-value pairs separated by commas. Used only on OS/390 platform. Optional. Valid attributes are:<br>• recordlength<br>• blocksize<br>• recordformat<br>• datasetorganization<br><br>**Note:** More DCB attributes are available but only these are supported in Sterling Integrator. Any other attribute will cause the copy to fail.For more information, see the Connect:Direct OS/390 user documentation. |
| LocalDocumentName | Name of the document in Sterling Integrator to which the remote file should be copied. Optional. |

| Field | Description |
|---|---|
| LocalFromCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:<br><br>• Code page specified is supported by the local Java VM<br>• Parameter LocalToCodePage is defined<br>• Text mode file transfer (BinaryMode = No)<br><br>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:<br><br><assign to="LocalFromCodePage">Cp1251</assign><br><br><assign to="LocalToCodePage">Cp1047</assign>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalStripBlanks | Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default). |
| LocalToCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:<br><br>• Code page specified is supported by the local Java VM<br>• Parameter LocalFromCodePage is defined<br>• Text mode file transfer (BinaryMode = No)<br><br>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:<br><br><assign to="LocalFromCodePage">Cp1251</assign><br><br><assign to="LocalToCodePage">Cp1047</assign>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalXlate | Specifies if the Connect:Direct Server adapter is to perform EBCDIC to ASCII translation before creating a new document from the received remote file . This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No. |
| RemoteFileName | Name of the file that should be copied from the remote Connect:Direct server. The path to the file should also be included in the name, if required by the remote node. Required. Enter the legal characters in the file name. |

| Field | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| Sysopts | Remote system operations to be implemented on the remote Connect:Direct node. Optional. For example (for copying to Connect:Direct Windows only):<br><br>Datatype(text) xlate(yes)For more information, see your Connect:Direct documentation. |

## Parameters Passed from Service to Business Process

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

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

## Business Process Example

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

```
<process name=" CopyFromExample">
  <sequence name="CDServerCopyFromGIS">
    <!-- Begin Server Session -->
        <operation name="CONNECT:Direct Server Begin Session Service">
          <participant name="CDServerBeginSession"/>
          <output message="BeginSession">
            <assign to="." from="*"></assign>
            <assign to="LocalCDNodeName">TEST1_CDSERVER</assign>
            <assign to="RemoteCDNodeName">REMOTECDNAME1</assign>
```

```
                        <assign to="RemoteUserId">userid1</assign>
                        <assign to="RemotePasswd">password1</assign>
                  </output>
                  <input message="inmsg">
                    <assign to="BeginSessionResults" from="*"></assign>
                  </input>
              </operation>
        <!-- Perform Copy (Pnode-INbound) -->
              <operation name="CONNECT:Direct Server Copy From Service">
                  <participant name="CDServerCopyFrom"/>
                  <output message="CopyFrom">
                    <assign to="." from="*"></assign>
                    <assign to="SessionToken"
                            from="//BeginSessionResults/SessionToken/node()">
</assign>
                    <assign to="RemoteFileName">samplecopyfrom.txt</assign>
                    <assign to="LocalDocumentName">samplelocal</assign>
                    <assign to="BinaryMode">no</assign>
                    <assign to="Sysopts">datatype(text) xlate(yes)</assign>
                  </output>
                  <input message="inmsg">
                    <assign to="CopyFromResults" from="*"></assign>
                  </input>
              </operation>
        <!-- End Server Session -->
              <operation name="CONNECT:Direct Server End Session Service">
                  <participant name="CDServerEndSession"/>
                  <output message="EndSession">
                    <assign to="." from="*"></assign>
                    <assign to="SessionToken"
                            from="//BeginSessionResults/SessionToken/node()">
</assign>
                  </output>
                  <input message="inmsg">
                    <assign to="EndSessionResults" from="*"></assign>
                  </input>
              </operation>
          </sequence>
        </process>
```

# Chapter 28. Connect:Direct Server CopyTo Service

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

| System Name | Connect:Direct Server CopyTo Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect Direct |
| Description | This service copies a file to a remote Connect:Direct server. To confirm that the copy performed by this service can restart if a Sterling Integrator crash interrupts the copy operation, the business process (in which this service is used) must have the recovery level set to Auto-resume. If so marked, the business process and the copy are resumed automatically when Sterling Integrator restarts, if the remote Connect:Direct node is still active. When resumed, the copy is restarted from the last checkpoint. |
| Business usage | A business user would use this service to copy a file to a remote Connect:Direct. |
| Usage example | A trading partner copies a business document to a remote Connect:Direct server outside the organization. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | Whenever the Connect:Direct Server CopyTo service is used in a business process, the Connect:Direct Server Primitive CopyTo service is exercised behind the scenes as part of a system business process to perform the actual copy operation. The Connect:Direct Server CopyTo service has tight integration with the Connect:Direct Server Begin Session service. A session must be established between two Connect:Direct nodes using the Begin Session service in order to execute the CopyTo service successfully. The Connect:Direct Server End Session service must be used to end the session established between the Connect:Direct nodes after the Connect:Direct Server CopyTo service. |
| Application requirements | Application requirements:<br>• Perimeter server<br>• Connect:Direct Windows 4.1.00 or later<br>• Connect:Direct UNIX 3.5.00 or later<br>• Connect:Direct OS/390 4.4.00 or later<br>• Connect:Direct OS/400 3.5.00 or later |
| Initiates business processes? | No |
| Invocation | A user who has permission to perform this activity must start the business process that runs this service. |

| Business process context considerations | LocalStripBlanks and LocalXlate are optional parameters. Use these parameters only when you want to control non-standard behavior and you have a complete understanding of your environment, the local content, the target requirement, data translation, and character set conversion. If LocalStripBlanks and LocalXlate are not explicitly assigned in the business process (see example below), translation occurs automatically during transmission. Example of a BP that does not explicitly assign LocalStripBlanks and LocalXlate parameters: |
|---|---|
| | `<operation name="CD Server Copy To Service">`<br>`<participant`<br>`name="ConnectDirectServerCopyToService001"/>`<br>`<output message="CopyTo">`<br>`    <assign to="." from="*">`<br>`</assign>`<br>`<assign`<br>`to="RemoteFileName">purchase-order.txt</assign>`<br>`</output>`<br>`    <input message="inmsg">`<br>`<assign to="CopyToResults"`<br>`from="*"></assign>      </input>`<br>`</operation>` |
| | If LocalStripBlanks and LocalXlate are explicitly assigned in the business process (see example below), the business process controls the transfer and the specified values are used.<br>**Note:** If the LocalXlate parameter is set to 'Yes', the SysOpts parameter must also specify 'xlate (yes)'.Example of a BP that explicitly assigns LocalStripBlanks and LocalXlate parameters: `<operation name="CD Server Copy To Service"> <participant name="ConnectDirectServerCopyToService001"/> <output message="CopyTo"> <assign to="." from="*"></assign> <assign to="BinaryMode">NO</assign> <assign to="LocalXlate">NO</assign> <assign to="LocalStripBlanks">NO</assign> <assign to="RemoteFileName">purchase-order.txt</assign> </output> <input message="inmsg"> <assign to="CopyToResults" from="*"></assign> </input> </operation>` |

| Returned status values | Returned status values: |
|---|---|
| | • Success – Normal execution with the return parameters specified below. |
| | Advanced Errors: |
| | • Invalid Document Error – Business process does not have document to be copied to a target Connect:Direct server. |
| | • CopyTo Remote File Error – An error related to file on remote Connect:Direct server. Might the file error or file path is invalid or invalid file permissions. |
| | • Invalid Sysopts Error – Sysopts might contain Invalid characters |
| | • Connect:Direct Server Error – Generic error associated with the Connect:Direct server has occurred. |
| | • Connect:Direct Error – Generic error associated with the Connect:Direct server has occurred. |
| | • Connect:Direct Service Error – Generic error associated with this service has occurred. |
| Restrictions | None |
| Persistence level | None |
| Testing considerations | Debug information for this service can be found in the Connect:Direct Interop log files. |

## Implementing the Connect:Direct Server CopyTo Service

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

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

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

## Configuring the Connect:Direct Server CopyTo Service

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| BinaryMode | Specifies the mode for Connect:Direct File transfer. Optional. Valid values are: |
| | • Yes - simple binary mode |
| | • No - (default) - text mode |
| | • VB - a variable block binary transfer with Connect:Direct on an OS/390 |

| Field | Description |
|---|---|
| CheckpointInterval | Specifies the byte interval for checkpoint support, which allows restart of interrupted transmissions at the last valid checkpoint point, reducing the time it takes to retransmit the file. Optional. Valid value is number of bytes, in the format: *nnnnnnnnnn*. Sterling Integrator stores checkpoint information for 30 days, after which it is automatically purged. If your database is corrupted, purge this checkpoint information by performing a cold restart. To do this, edit the adapter configuration Server Start Option parameter to Cold Start. Once you have performed the cold start, edit the adapter configuration to change the setting back to Warm Start. You can also purge the checkpoint information from the database manually. |
| Compression | Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9. |
| DCB | Specifies Data Control Block (DCB), a group of attributes specified as name-value pairs separated by commas. One, some, or all values may be included. Pattern is name=value with comma between, as in the following example: <br><br> recordlength=200, blocksize=0, recordformat=FB, datasetorganization=PS, unittype=cart, expirationdate=06365, volumecount=99Used only on OS/390 platform. Optional. Valid attributes are: <br> • recordlength <br> • blocksize <br> • recordformat <br> • datasetorganization <br> • expirationdate <br> • volumecount <br> • unittype <br><br> **Note:** More DCB attributes are available but only these are supported in Sterling Integrator. Any other attribute will cause the copy to fail.For more information, see the Connect:Direct for OS/390 user documentation. |
| Disposition | Specifies how the file gets created on the remote Connect:Direct server. Optional. Valid values are: <br> • New – Specifies that the CopyTo service will create the destination file. The copy fails if the file exists <br> • rpl – Specifies that the destination file name replaces any existing file or, if none exists, allocates a new file |
| LocalDocumentId | A valid document ID for an existing document in Sterling Integrator. When this field is left blank, the primary document provided to the business process is used. Optional. |
| LocalFileNameLabel | Specifies a label for the name of the local file. This file name label is provided to the remote Connect:Direct node for informational purposes only. Optional. |

| Field | Description |
|---|---|
| LocalFromCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:<br>• Code page specified is supported by the local Java VM<br>• Parameter LocalToCodePage is defined<br>• Text mode file transfer (BinaryMode =No)<br><br>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:<br><br><assign to="LocalFromCodePage">Cp1251</assign><br><br><assign to="LocalToCodePage">Cp1047</assign>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalStripBlanks | Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default). |
| LocalToCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:<br>• Code page specified is supported by the local Java VM<br>• Parameter LocalFromCodePage is defined<br>• Text mode file transfer (BinaryMode =No)<br><br>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:<br><br><assign to="LocalFromCodePage">Cp1251</assign><br><br><assign to="LocalToCodePage">Cp1047</assign>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalXlate | Specifies if the Connect:Direct Server adapter is to perform ASCII to EBCDIC translation before copying a file to a remote Connect:Direct Server. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No. |
| CDRecordType | Identifies to the remote Connect:Direct which class to process the service. For example:<br><br>1Optional. For more information, see the Connect:Direct for OS/390 user documentation. |
| RemoteFileName | Name that should be given to the file that is copied to the remote Connect:Direct server. Required. |

| Field | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| Sysopts | Remote system operations to be implemented on the remote Connect:Direct node. Optional. For example (for copying to Connect:Direct Windows only):<br><br>Datatype(text) xlate(yes)For more information, see your Connect:Direct documentation. |

## Parameters Passed from Business Process to Service

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

| Field | Description |
|---|---|
| BinaryMode | Specifies the mode for Connect Direct File transfer. Optional. Valid values are:<br>• Yes - simple binary mode<br>• No - (default) - text mode<br>• VB - a variable block binary transfer with Connect:Direct on an OS/390 |
| CheckpointInterval | Specifies the byte interval for checkpoint support, which enables restart of interrupted transmissions at the last valid checkpoint point, reducing the time it takes to retransmit the file. Optional. Valid value is number of bytes, in the format: *nnnnnnnnnn*. |
| Compression | Specifies that the data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values are 1-9. |

| Field | Description |
|-------|-------------|
| DCB | Specifies Data Control Block (DCB), a group of attributes specified as name-value pairs separated by commas. One, some, or all values may be included. Pattern is name=value with comma between, as in the following example:recordlength=200, blocksize=0, recordformat=FB, datasetorganization=PS, unittype=cart, expirationdate=06365, volumecount=99Used only on OS/390 platform. Optional. Valid attributes are:<br><br>• recordlength<br><br>• blocksize<br><br>• recordformat<br><br>• datasetorganization<br><br>• expirationdate<br><br>• volumecount<br><br>• unittype<br><br>**Note:** More DCB attributes are available but only these are supported in Sterling Integrator. Any other attribute will cause the copy to fail.For more information, see the Connect:Direct for OS/390 user documentation. |
| Disposition | Specifies how the file gets created on the remote Connect:Direct server. Optional. Valid values are:<br><br>• New – Specifies that the CopyTo service will create the destination file. The copy fails if the file exists<br><br>• rpl – Specifies that the destination file name replaces any existing file or, if none exists, allocates a new file |
| LocalDocumentId | A valid document ID for an existing document in Sterling Integrator. Optional. |
| LocalFileNameLabel | Specifies a label for the name of the local file. This file name label is provided to the remote Connect:Direct node for informational purposes only. Optional. |
| LocalFromCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalToCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:<br><br>• Code page specified is supported by the local Java VM<br><br>• Parameter LocalToCodePage is defined<br><br>• Text mode file transfer (BinaryMode = No)<br><br>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:<br><br><assign to="LocalFromCodePage">Cp1251</assign><br><br><assign to="LocalToCodePage">Cp1047</assign>This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalStripBlanks | Specifies whether the Connect:Direct Server adapter will strip blanks from the end of records in a received remote file before creating a new document. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes or No (default). |

| Field | Description |
|---|---|
| LocalToCodePage | Specifies a code page supported by Java VM. Used in conjunction with the value of the LocalFromCodePage parameter. The Connect:Direct Server adapter will apply a code page conversion to a document before transferring it to a remote Connect:Direct Server. This parameter is only valid with the following conditions:<br>• Code page specified is supported by the local Java VM<br>• Parameter LocalFromCodePage is defined<br>• Text mode file transfer (BinaryMode = No)<br>Optional. Valid value is the name of a codepage supported by the local Java VM. For example:<br>&lt;assign to="LocalFromCodePage"&gt;Cp1251&lt;/assign&gt;<br>&lt;assign to="LocalToCodePage"&gt;Cp1047&lt;/assign&gt;This example will translate the local file from windows latin-1 to ibm ebcdic. See http://java.sun.com for a complete list of supported encodings. |
| LocalXlate | Specifies if the Connect:Direct Server adapter is to perform ASCII to EBCDIC translation before copying a file to a remote Connect:Direct Server. This option is only valid for Text mode file transfers (BinaryMode = No). Optional. Valid value is Yes (default) or No. |
| RemoteFileName | Name that should be given to the file that is copied to the remote Connect:Direct server. Required. |
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| Sysopts | Remote system operations to be implemented on the remote Connect:Direct node. Optional. For example (for copying to Connect:Direct Windows only):<br>Datatype(text) xlate(yes)For more information, see your Connect:Direct documentation. |

## Parameters Passed from Service to Business Process

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

| Parameter | Description |
|---|---|
| LocalCDNodeName | Local Connect:Direct server instance name. Required. |

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

## Business Process Example

The following example illustrates using the Connect:Direct Server CopyTo service in a business process. The business process copies the current primary document to the remote Connect:Direct server, which is identified with a session token.

```
<process name=" CopyToExample">
  <sequence name="CDServerCopyToGIS">
    <!-- Begin Server Session -->
        <operation name="CONNECT:Direct Server Begin Session Service">
          <participant name="CDServerBeginSession"/>
          <output message="BeginSession">
            <assign to="." from="*"></assign>
            <assign to="LocalCDNodeName">TEST1_CDSERVER</assign>
            <assign to="RemoteCDNodeName">REMOTECDNODE1</assign>
            <assign to="RemoteUserId">user1</assign>
            <assign to="RemotePasswd">password1</assign>
          </output>
          <input message="inmsg">
            <assign to="BeginSessionResults" from="*"></assign>
          </input>
        </operation>
        <operation name="CONNECT:Direct Server Copy To Service">
          <participant name="CDServerCopyTo"/>
          <output message="CopyTo">
            <assign to="." from="*"></assign>
            <assign to="SessionToken"
                    from="//BeginSessionResults/SessionToken/node()">
</assign>
            <assign to="RemoteFileName">sample.txt</assign>
            <assign to="BinaryMode">no</assign>
            <assign to="Sysopts">datatype(text) xlate(no)</assign>
          </output>
          <input message="inmsg">
            <assign to="CopyToResults" from="*"></assign>
          </input>
        </operation>
    <!-- End Server Session -->
        <operation name="CONNECT:Direct Server End Session Service">
          <participant name="CDServerEndSession"/>
          <output message="EndSession">
            <assign to="." from="*"></assign>
            <assign to="SessionToken"
                    from="//BeginSessionResults/SessionToken/node()">
</assign>
          </output>
          <input message="inmsg">
            <assign to="EndSessionResults" from="*"></assign>
```

```
                  </input>
              </operation>
          </sequence>
      </process>
```

# Chapter 29. Connect:Direct Server End Session Service

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

| System Name | Connect:Direct Server End Session Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect:Direct |
| Description | This service is used to end a session established between two Connect:Direct nodes. |
| Business usage | A business user would use this service to terminate a session with a remote Connect:Direct Node. The service uses the Connect:Direct Server adapter as the communications mechanism for the outbound request to end the session. |
| Usage example | A Sterling Integrator business process initiates a session with a remote Connect:Direct node using the Connect:Direct Begin Session service and the Connect:Direct Server adapter. During the course of the session, business activity occurs: for example, a file is copied from the remote Connect:Direct node. After the operation is complete, the business process uses the Connect:Direct End Session service to terminate the session. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | Related services:<br><br>• Connect:Direct Server Begin Session service – used to initiate the session that will be terminated by this service.<br><br>• Connect:Direct Server adapter – The Connect:Direct End Session service uses a Connect:Direct Server adapter to communicate with a remote Connect:Direct server. |
| Application requirements | Application requirements:<br><br>• Perimeter server<br><br>• Connect:Direct Windows 4.1.00 or later<br><br>• Connect:Direct UNIX 3.5.00 or later<br><br>• Connect:Direct OS/390 4.4.00 or later<br><br>• Connect:Direct OS/400 3.5.00 or later |
| Initiates business processes? | No |
| Invocation | This service is run by a business process to terminate a session with a remote Connect:Direct node. |
| Business process context considerations | None |

| Returned status values | Returned status values: |
|---|---|
| | • Success – Normal execution with the return parameters specified below. |
| | • Invalid Session Token – The Session Token specified to identify the Session was not valid. |
| | • Connect:Direct Server Error – A generic error associated with the Connect:Direct Server has occurred. |
| | • Connect:Direct Error – A generic error associated with Connect:Direct has occurred. |
| | • Connect:Direct Service Error – A generic error associated with this service has occurred. |
| Restrictions | You must use this service with a Connect:Direct Begin Session service. The session token information in process data is necessary to configure the service. |
| Persistence level | None |
| Testing considerations | Debug information for this service can be found in the Connect:Direct Interop log files. |

## Implementing the Connect:Direct Server End Session Service

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

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

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

## Configuring the Connect:Direct Server End Session Service

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |

| Field | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |

## Parameters Passed from Business Process to Service

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

| Field | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |

## Parameters Passed from Service to Business Process

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

| Parameter | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br><br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| SessionEndTime | Specifies the date time when the session was ended. Required. |

## Business Process Example

The following example illustrates using the Connect:Direct Server End Session service in a business process:

```
<process name="CDInterop_SessionBeginAndEnd">
  <sequence name="CDServerSessionBeginGIS">
    <!-- Begin Server Session -->
      <operation name="CONNECT:Direct Server Begin Session Service">
        <participant name="CDServerBeginSession"/>
        <output message="BeginSession">
          <assign to="." from="*"></assign>
          <assign to="LocalCDNodeName">TEST1_CDSERVER</assign>
          <assign to="RemoteCDNodeName">NMAYAKUNTLAVM2K</assign>
          <assign to="RemoteUserId">ksticken</assign>
          <assign to="RemotePasswd">kirk123</assign>
       </output>
        <input message="inmsg">
          <assign to="BeginSessionResults" from="*"></assign>
        </input>
      </operation>
    <!-- End Server Session -->
      <operation name="Connect:Direct Server End Session Service">
        <participant name="CDServerEndSession"/>
        <output message="EndSession">
          <assign to="." from="*"></assign>
          <assign to="SessionToken"
      from="//BeginSessionResults/SessionToken/node()"></assign>
        </output>
        <input message="inmsg">
          <assign to="EndSessionResults" from="*"></assign>
        </input>
      </operation>
  </sequence>
</process>
```

# Chapter 30. Connect:Direct Server Run Job Service

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

| System Name | Connect:Direct Server Run Job Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect Direct |
| Description | This service submits work to the host operating system that executes asynchronously to the remaining steps in a process. The work can execute on either the local or remote node. |
| Business usage | To execute a job on a mainframe, write a business process that uses the Run Job service to execute a script on a remote Connect:Direct node on 0S/390. |
| Usage example | Executing a script asynchronously on a remote Connect:Direct server that compresses and archives log files. |
| Preconfigured? | No. A Connect:Direct Server adapter must be configured before using this service. |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | The Connect:Direct Server Run Job service is designed to work surrounded by a Connect:Direct Server Begin Session service and a Connect:Direct Server End Session service. It uses the Connect:Direct Server adapter to communicate with remote Connect:Direct servers. |
| Application requirements | Application requirements:<br>• Perimeter server<br>• Connect:Direct Windows 4.1.00 or later<br>• Connect:Direct UNIX 3.5.00 or later<br>• Connect:Direct OS/390 4.4.00 or later<br>• Connect:Direct OS/400 3.5.00 or later |
| Initiates business processes? | No |
| Invocation | The business process running this service must be started by a user with permission to issue the Run Job statement on the target Connect:Direct server. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• Success – Normal execution with the return parameters specified below.<br>• File not found error – Specified file was not found.<br>• Access to file denied error – Specified file could not be accessed.<br>• Unable to start file error – Specified file could not be started. |
| Restrictions | Connect:Direct nodes on iSeries do not support System Options (Sysopts), an optional parameter for this service. |

| | |
|---|---|
| Persistence level | None |
| Testing considerations | Debug information for this service can be found in the Connect:Direct Interop log files. |

## Implementing the Connect:Direct Server Run Job Service

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

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

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

## Configuring the Connect:Direct Server Run Job Service

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| SessionToken | Specifies the identifier for the session established between two Connect:Direct server nodes. Required. |
| Sysopts | Specifies the parameters appropriate for the operation being performed. Required. Valid values are:<br>• pgm(*filespec*) specifies which .EXE or .BAT file to run.<br>• cmd(*command* \| *parms*) specifies a system command and any arguments that this command requires.<br>• args(*arguments*) specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify pgm. |

## Parameters Passed from Business Process to Service

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

| Parameter | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| Sysopts | Specifies the parameters appropriate for the operation being performed. Required. Valid values are:<br>• pgm(*filespec*) specifies which .EXE or .BAT file to run.<br>• cmd(*command | parms*) specifies a system command and any arguments that this command requires.<br>• args(*arguments*) specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify pgm. |

## Parameters Passed from Service to Business Process

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

| Parameter | Description |
|---|---|
| LocalCDNodeName | The Sterling Integrator Connect:Direct Server adapter name. |
| RemoteCDNodeName | The remote Connect:Direct server IP address. |
| Status | Indicates whether or not the remote job was executed. **Note:** This does not provide address the success or failure of the executed job, only that it was executed. |

## Business Process Examples

The following examples illustrate using the Connect:Direct Server Run Job service in a UNIX environment and a Windows environment:

The following is a UNIX example:

```
<process name="RunJobServiceExample">
  <sequence>
      <operation name="CONNECT:Direct Server Begin Session Service">
        <participant name="CDServerBeginSession"/>
        <output message="BeginSession">
          <assign to="." from="*"></assign>
          <assign to="LocalCDNodeName">CDSERVER_ADAPTER_TEST1</assign>
          <assign to="RemoteCDNodeName">CDSERVER_ADAPTER_TEST2</assign>
```

```
                              <assign to="RemoteUserId">USERID</assign>
                              <assign to="RemotePasswd">PASSWORD</assign>
                              <assign to="PlexClass">ABC</assign>
                          </output>
                          <input message="inmsg">
                  <assign to="BeginSessionResults" from="*"></assign>
                          </input>
                      </operation>
                      <operation name="Connect:Direct Server Run Job Service">
                          <participant name="CDServerRunJob"/>
                          <output message="RunJob">
                              <assign to="." from="*"></assign>
                              <assign to="SessionToken"
                  from="//BeginSessionResults/SessionToken/node()"></assign>
                              <assign to="Sysopts" from="'cd /home/sterlingcommerce/script; sh
                                    runScript.sh argument1'"></assign>
                          </output>
                          <input message="inmsg">
                              <assign to="RunJob_Results" from="*"></assign>
                          </input>
                      </operation>
                      <operation name="CONNECT:Direct Server End Session Service">
                          <participant name="CDServerEndSession"/>
                          <output message="EndSession">
                              <assign to="." from="*"></assign>
                              <assign to="SessionToken"
                                      from="//BeginSessionResults/SessionToken/node()"></assign>
                          </output>
                          <input message="inmsg">
                              <assign to="EndSessionResults" from="*"></assign>
                          </input>
                      </operation>
                  </sequence>
              </process>
```

The following is a Windows example:

```
<process name="RunJobServiceExample">
  <sequence>
          <operation name="CONNECT:Direct Server Begin Session Service">
              <participant name="CDServerBeginSession"/>
              <output message="BeginSession">
                  <assign to="." from="*"></assign>
                  <assign to="LocalCDNodeName">CDSERVER_ADAPTER_TEST1</assign>
                  <assign to="RemoteCDNodeName">CDSERVER_ADAPTER_TEST2</assign>
                  <assign to="RemoteUserId">USERID</assign>
                  <assign to="RemotePasswd">PASSWORD</assign>
                  <assign to="PlexClass">ABC</assign>
              </output>
              <input message="inmsg">
                  <assign to="BeginSessionResults" from="*"></assign>
              </input>
          </operation>
          <operation name="Connect:Direct Server Run Job Service">
              <participant name="CDServerRunJob"/>
              <output message="RunJob">
                  <assign to="." from="*"></assign>
                  <assign to="SessionToken"
          from="//BeginSessionResults/SessionToken/node()"></assign>
                  <assign to="Sysopts"
          from="'pgm(D:\SterlingProject\CDInterop\Demo\test.bat)
            args(TEST)'"></assign>
              </output>
              <input message="inmsg">
                  <assign to="RunJob_Results" from="*"></assign>
              </input>
          </operation>
```

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

# Chapter 31. Connect:Direct Server Run Task Service

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

| System Name | Connect:Direct Server Run Task Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect Direct |
| Description | This service executes a program or command synchronously to the remaining steps in a Process. The program or command can execute locally or on the remote Connect:Direct node. User programs can be run in different environments. |
| Business usage | You want to execute a job on a mainframe. You can write a business process that uses the Run Task service to execute a script on a remote Connect:Direct node on 0S/390 and then wait for the completion of the run task operation. |
| Usage example | You execute a script that compresses and archives log files synchronously on a remote Connect:Direct Server. |
| Preconfigured? | A Connect:Direct Server adapter must be configured before using this service |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The Connect:Direct Server Run Task Service works between a Connect:Direct Server Begin Session Service and a Connect:Direct Server End Session service. |
| Application requirements | Application requirements:<br>• Perimeter server<br>• Connect:Direct Windows 4.1.00 or later<br>• Connect:Direct UNIX 3.5.00 or later<br>• Connect:Direct OS/390 4.4.00 or later<br>• Connect:Direct OS/400 3.5.00 or later |
| Initiates business processes? | No |
| Invocation | The business process running this service must be started by a user with permission to issue the Run Task statement on the target Connect:Direct server. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• Success – Normal execution with the return parameters specified below.<br>• File not found Error – Specified file was not found.<br>• Access to file denied Error – Specified file could not be accessed.<br>• Unable to start file Error – Specified file could not be started.<br>• Program Failure Error – Failure of the program to be executed by the specified file. |

| Restrictions | Connect:Direct nodes on iSeries do not support System Options (Sysopts), an optional parameter for this service. |
|---|---|
| Persistence level | None |
| Testing considerations | None |

## Implementing the Connect:Direct Server Run Task Service

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

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

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

## Configuring the Connect:Direct Server Run Task Service

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| Program | Specifies the parameters appropriate for the operation being performed. Required.<br>**Note:** In Sterling Integrator Release 3.1, the name of this parameter was System Options.The valid values depend on the destination Connect:Direct server. For example, the valid values for Connect:Direct Windows are:<br>• pgm (*filespec*) specifies which .exe or .bat file to run.<br>• cmd (*command \| parms*) specifies a system command and any arguments that this command requires.<br>• args (*arguments*) specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify pgm. |

| Field | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| SystemOpts | Specifies the arguments passed to the program when it is started. Optional.<br>**Note:** Connect:Direct AS/400 nodes do not support SystemOpts. |

## Parameters Passed from Business Process to Service

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

| Field | Description |
|---|---|
| Program | Specifies the parameters appropriate for the operation being performed. Required.<br>**Note:** In Sterling Integrator Release 3.1, the name of this parameter was System Options. The valid values depend on the destination Connect:Direct server. For example, the valid values for Connect:Direct Windows are:<br>• pgm (*filespec*) specifies which .exe or .bat file to run.<br>• cmd (*command* \| *parms*) specifies a system command and any arguments that this command requires.<br>• args (*arguments*) specifies the arguments passed to the program when it is started. These arguments are in the same format specified from the command prompt. This optional parameter is only valid when you specify pgm. |

| Field | Description |
|---|---|
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |
| SystemOpts | Specifies the arguments passed to the program when it is started. Optional.<br>**Note:** Connect:Direct AS/400 nodes do not support SystemOpts. |

## Parameters Passed from Service to Business Process

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

| Parameter | Description |
|---|---|
| LocalCDNodeName | The Sterling Integrator Connect:Direct Server adapter name |
| RemoteCDNodeName | The remote Connect:Direct server IP address |
| Status | Indicates whether or not the remote job was executed.<br>**Note:** This does not address the success or failure of the executed job, only that it was executed. |

## Business Process Example for Connect:Direct for UNIX

Each parameter's value and syntax is defined by the remote Connect:Direct platform and underlying operating system. When the remote platform operating system is UNIX/Linux, the command and its arguments, if any, are specified as the Program parameter value and in the same syntax that they would be entered at a shell prompt. For example, to execute the 'ls' command with the '-1' option (single column output) and send the output to a file called 'ls.out' locally on a UNIX/Linux host, type the following at a shell prompt:

```
$ ls -1 > ls.out
```

To specify this operation in the RunTask Service, assign the text to the RunTask Service Program parameter in the business process:

```
<assign to="Program">ls -1 /tmp &gt; ls.out</assign>
```

**Note:** The redirection operator (>) must be encoded (&gt;) to prevent XML processing errors.

The following is a business process for UNIX with the Program parameter:

```
<process name = "SampleRunTask">
  <sequence>
    <operation name="CD Server Begin Session Service">
...
    </operation>
    <operation name="CD Server Run Task Service">
      <participant name="CDServerRunTask"/>
      <output message="CDServerRunTaskServiceTypeInputMessage">
        <assign to="Program">ls -1 /tmp &gt; ls.out</assign>
 <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="CD Server End Session Service">
...
    </operation>
  </sequence>
</process>
```

## Business Process Example for Connect:Direct for Windows

When a Connect:Direct Windows node receives a request to execute a task, the business process must specify whether or not the task requires a cmd shell to execute. Connect:Direct Windows defines a syntax for communicating this information. To execute a program, the pgm and args keywords are used:

```
<assign to="Program">pgm(\path\file.exe) args(arg1 arg2 arg3)</assign>
```

**Note:** The args keyword is only used when arguments are needed. The path is computed according to Connect:Direct rules. If the Program directory is defined for the submitting user on the remote node, the pgm value will be appended to its value prior to use.

To execute a Windows command:

```
<assign to="Program">cmd(command arg1 arg2 arg3)</assign>
```

The following business process demonstrates how to configure the Connect:Direct RunTask Service:

```
<process name = "SampleRunTask">
  <sequence>
    <operation name="CD Server Begin Session Service">
...
    </operation>
    <operation name="CD Server Run Task Service">
      <participant name="CDServerRunTask"/>
      <output message="CDServerRunTaskServiceTypeInputMessage">
<assign to="Program">cmd(dir \tmp &gt; \tmp\dir.out)</assign>
</output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="CD Server End Session Service">
...
    </operation>
  </sequence>
</process>
```

## Parameter for Connect:Direct for Z/OS

The SystemOpts parameter value is appended to the Program parameter value. Use this parameter when the remote node is Connect:Direct for Z/OS. In this case, the Program parameter value specifies the remote program name only. The remote program parameters, if any, are specified using the SystemOpts parameter.

# Chapter 32. Connect:Direct Server Submit Service

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

| System Name | Connect:Direct Server Submit Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect Direct |
| Description | This service is used to submit a Connect:Direct process to another Connect:Direct node. The submitted process must reside in a file on the destination Connect:Direct node. |
| Business usage | A business user would use this service to initiate work on a remote Connect:Direct node. This service performs the equivalent of a Submit statement in a Connect:Direct process. During execution of a Connect:Direct process, the Submit statement causes another Connect:Direct process to be submitted to either the Pnode, which is the node with process control, or the Snode, which is the remote node that participates in the process. Any parameter values that are specified in the Submit statement override the parameter values contained in a process statement that is named by the file parameter. |
| Usage example | A business user has a file that exists on a remote node where Connect:Direct is running. This file is needed on another node. The user creates a business process that submits a Connect:Direct process to a remote Connect:Direct server to transfer the file from one node to the other. |
| Preconfigured? | A Connect:Direct Server adapter must be configured before using this service. |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Connect:Direct Server adapter |
| Application requirements | Application requirements: <br> • Perimeter server <br> • Connect:Direct Windows 4.1.00 or later <br> • Connect:Direct UNIX 3.5.00 or later <br> • Connect:Direct OS/390 4.4.00 or later <br> • Connect:Direct OS/400 3.5.00 or later |
| Initiates business processes? | No |
| Invocation | The business process running this service must be started by a user with permission to issue the Submit statement on the target Connect:Direct server. |
| Business process context considerations | None |

| Returned status values | Returned status values: |
|---|---|
| | • Invalid Business Process Error – User tried to submit an invalid business process in Sterling Integrator. |
| | • User Permission Error – User does not have permission to submit the business process in Sterling Integrator. |
| | • Connect:Direct Server Error – A generic error associated with the Connect:Direct Server has occurred. |
| | • Connect:Direct Error – A generic error associated with Connect:Direct has occurred. |
| | • Connect:Direct Service Error – A generic error associated with the Connect:Direct service has occurred. |
| Restrictions | If the Submit Process request is issued to a remote Connect:Direct node that is operating on OS/400, the request will fail. The OS/400 Connect:Direct node does not support Submit Process. |
| Persistence level | None |
| Testing considerations | Debug information for this service can be found in the Connect:Direct Interop log files. |

## Implementing the Connect:Direct Server Submit Service

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

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

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

## Configuring the Connect:Direct Server Submit Service

To configure the Connect:Direct Server Submit service, you must define the fields in Sterling Integrator and in the GPM.

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| RemoteProcessFile | Process file on the remote Connect:Direct server. This is the process that is submitted. Required. |
| RemoteProcessParms | The parameters or the arguments that can be passed dynamically to the process file on remote Connect:Direct server. Optional. UNIX Example: SNODE=10.20.80.117;1364 |

| Field | Description |
|---|---|
| SubmitUserId | User ID for remote system other than the remoteUserId specified in the Connect:Direct Server Begin Session service. Optional. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service). |
| SubmitPasswd | Password for remote system for user ID specified in SubmitUserId. Required when SubmitUserId is specified. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses password associated with the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service) |
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements: <br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |

## Parameters Passed from Business Process to Service

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

| Parameter | Description |
|---|---|
| RemoteProcessFile | Process file on the remote Connect:Direct server. This is the process that is submitted. Required. |
| RemoteProcessParms | The parameters or the arguments that can be passed dynamically to the process file on remote Connect:Direct server. Optional. UNIX Example: SNODE=10.20.80.117;1364 |
| SubmitUserId | User ID for remote system other than the remoteUserId specified in the Connect:Direct Server Begin Session service. Optional. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service). |

| Parameter | Description |
|---|---|
| SubmitPasswd | Password for remote system for user ID specified in SubmitUserId. Required when SubmitUserId is specified. Valid value is any character string with a maximum of 30 characters. By default, the remote Connect:Direct server uses password associated with the user ID authenticated at the beginning of the current session (remoteUserId in the Begin Session service) |
| SessionToken | Identifies an established session between two Connect:Direct nodes. Required. Valid value is an XML structure that includes the following elements:<br>• SessionId<br>• ProcessNumber<br>• ProcessName<br>• SubmitDateTime<br>• RemoteCDNodeName<br>• RemoteUserId<br>• RemotePasswd<br>• PlexClass<br>• MaxRestartRetries<br>• RestartRetryInterval |

## Parameters Passed From Service to Business Process

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

| Parameter | Description |
|---|---|
| LocalCDNodeName | Specifies the local Connect:Direct which submitted the process. Required. |
| RemoteCDNodeName | Specified the remote Connect:Direct on which the process was submitted. Required. |
| Status | Specifies the status of the submit statement execution. Required. Valid values are SUCCESS and ERROR. |
| SubmittedProcessNum | Specifies the process number assigned to the process. Optionally returned (if > –1). |
| SubmittedProcessDateTime | Specifies the date and time when the submit statement is executed. Required. |

## Business Process Example

The following example illustrates using the Connect:Direct Server Submit service in a business process:

```
<process name=" SubmitStatementExample">
  <sequence name="CDServerSubmit">
    <!-- Begin Server Session -->
      <operation name="CONNECT:Direct Server Begin Session Service">
        <participant name="CDServerBeginSession"/>
        <output message="BeginSession">
          <assign to="." from="*"></assign>
          <assign to="LocalCDNodeName">TEST1_CDSERVER</assign>
          <assign to="RemoteCDNodeName">NMAYAKUNTLAVM2K</assign>
          <assign to="RemoteUserId">joeuser</assign>
```

```
              <assign to="RemotePasswd">joe123</assign>
            </output>
            <input message="inmsg">
              <assign to="BeginSessionResults" from="*"></assign>
            </input>
          </operation>
          <operation name="Connect:Direct Server Submit Service">
            <participant name="CDServerSubmit"/>
            <output message="SubmitRequest">
              <assign to="." from="*"></assign>
              <assign to="SessionToken"
                    from="//BeginSessionResults/SessionToken/node()">
</assign>
              <assign to="RemoteProcessFile">Lalitha.cd</assign>
              <assign to="RemoteProcessParms">SNODE=00.00.00.000;1364</assign>
                      </output>
            <input message="inmsg">
              <assign to="SubmitResults" from="*"></assign>
            </input>
          </operation>
      <!-- End Server Session -->
          <operation name="CONNECT:Direct Server End Session Service">
            <participant name="CDServerEndSession"/>
            <output message="EndSession">
              <assign to="." from="*"></assign>
              <assign to="SessionToken"
                    from="//BeginSessionResults/SessionToken/node()">
</assign>
            </output>
            <input message="inmsg">
              <assign to="EndSessionResults" from="*"></assign>
            </input>
          </operation>
</sequence>
</process>
```

# Chapter 33. CDATA Conversion Service

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

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

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

### How the CDATA Conversion Service Works

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

### Outbound

In CDATA_TO_PCDATA mode, Sterling Integrator converts any CDATA sections to PCDATA, and converts the five standard entity characters (&, <, >, ", and ') to their corresponding entity references (&amp; &lt; &gt; &quot; &apos;). For example, see the following fragment:

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

would be converted to:

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

### Inbound

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

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

would be converted to:

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

### Code Sample

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

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

## Implementing the CDATA Conversion Service

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

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

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

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

## Configuring the CDATA Conversion Service

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

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

# Chapter 34. Connect:Enterprise UNIX Server Add Service

The following table provides an overview of the Connect:Enterprise UNIX Server Add service:

| System name | Connect:Enterprise UNIX Server Add Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX |
| Description | This service works with the Connect:Enterprise UNIX Server adapter to add a batch to a Connect:Enterprise UNIX mailbox. |
| Business usage | If used in a business process that is initiated from a Connect:Enterprise UNIX schedule notification, this service adds the batch and logs the activity into Connect:Enterprise UNIX as part of the schedule notification session. If the Add service is used in business process that begins an unsolicited session, the activity is represented in Connect:Enterprise UNIX as remote business process account activity. |
| Usage example | You want to send a business document to a trading partner who has a Connect:Enterprise UNIX mailbox. For more information, see *How the Connect:Enterprise UNIX Server Add Service Works*. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Extract service, Batch Status service, Log service. |
| Application requirements | An active instance of Sterling Integrator and Connect:Enterprise UNIX v2.2 or later are required for this service to work. |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |

| | |
|---|---|
| Business process context considerations | The service gets the Connect:Enterprise UNIX Server adapter name from one of two places: |
| | • when run in response to a schedule notification the CEUServerAdapterInstanceName is automatically placed in the process data |
| | • when run by some other means, this parameters could come from a previous service, or it must be defined in this service's configuration. |
| | If the process data does not already contain the CEUSessionId, then this service uses the Connect:Enterprise UNIX-generated CEUSessionId. Included in process data (when run by a schedule notification) are Put Options including data format and process flags. The Add service can carry these Put Options forward to be used when the batch is added, or you can override these flags by specifying these parameters in the Add service. |
| Returned status values | Returned status values: |
| | • Success – Normal execution with the return parameters specified. |
| | • Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. |
| | • Invalid Session – Indicates an invalid session. |
| | • Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. |
| | • Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise Unix Server Adapter. |
| | • CEU Service Error – A generic error associated with the service has occurred. |
| Restrictions | None |
| Persistence level | All |
| Testing considerations | Testing considerations: |
| | 1. Successfully establish connection from Sterling Integrator to Connect:Enterprise UNIX through a Connect:Enterprise UNIX Server adapter. |
| | 2. Create a business process with the Add service. |
| | 3. Pass in a primary document and it should be added into Connect:Enterprise UNIX as remote connect activity. |
| | For more information, see the predefined business process, CEUInterop-DemoPOAdd. |

## How the Connect:Enterprise UNIX Server Add Service Works

The following describes what happens when this service is used to add a business document to a Connect:Enterprise UNIX mailbox as the result of an unsolicited remote connect.

1. The Sterling Integrator business process initiates a communication session with Connect:Enterprise UNIX through the Connect:Enterprise UNIX Server adapter. The adapter submits a user ID and password to gain access.

2. The adapter returns a CEUSessionId to the Begin Session service. The CEUSessionId is put in the process data.

3. Sterling Integrator adds a batch to a specific mailbox using the Connect:Enterprise UNIX Server Add service. This service works through the Connect:Enterprise UNIX Server adapter to accomplish the add. Included in the transfer are: CEUMailboxId, CEUBatchId, and the file being transferred.

4. Connect:Enterprise UNIX stores the data in the mailbox and returns a notification to the adapter indicating that the data was transferred successfully. The notification includes the batch number Connect:Enterprise UNIX assigned to the file and the creation date.

5. A flag ("J" flag) is set on the batch in Connect:Enterprise UNIX to indicate that Sterling Integrator added the batch.

6. Sterling Integrator sets the correlations between the business process that initiated the request and batch summary data from Connect:Enterprise UNIX. These include: CEUBatchId, CEUMailboxId, CEUBatchNumber, schedule definition name and CEUSessionId. These correlations allow subsequent searches for this information using the Advanced Search function.

## Implementing the Connect:Enterprise UNIX Server Add Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Sterling Integrator, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Add service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.

2. Configure the pre-installed copy of the service. For information, see *Configuring the Connect:Enterprise UNIX Server Add Service*.

3. Use the Connect:Enterprise UNIX Server adapter in a business process.

## Configuring the Connect:Enterprise UNIX Server Add Service

To configure the Connect:Enterprise UNIX Server Add service, you must specify field settings in Sterling Integrator and in the GPM.

### Sterling Integrator Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Add service:

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

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

| Field | Description |
|---|---|
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br><br>• None – You do not want to include this configuration in a group at this time (default)<br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br><br>• 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.<br><br>For more information about service groups see *Managing Services and Adapters*. |
| CEU Server Adapter Instance Name(CEUServerAdapterInstanceName) | The instance of the Connect:Enterprise UNIX adapter that the Add service should use. The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP. |

## GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Add service in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| CEUBatchId | The name to be assigned to the batch when it is placed in the Connect:Enterprise UNIX repository. The primary document name is used if no name is specified. |
| CEUDataFormat | Sets the data format flag: ASCII, BINARY, EBCDIC. In an auto connect scenario, Connect:Enterprise UNIX supplies this flag in the process data within the schedule definition: ACD_INFO/PutOptions. In a remote connect scenario, Connect:Enterprise UNIX can use the Put Option defined in the remote account (RSD). Optional.<br><br>• If not specified, Connect:Enterprise UNIX uses the Put Option defined in the schedule (ACD_INFO) or in the remote account (RSD).<br><br>• Default - C (Collected), J (BP protocol flag), Y (binary data format)<br><br>• If specified, overrides the Connect:Enterprise UNIX Put Option. |
| CEUMailboxId | The ID of the Connect:Enterprise UNIX mailbox that is to receive the document. The document is added to the mailbox for the account name if no CEUMailboxId is specified. |

| Field | Description |
|---|---|
| CEUProcessOption | Sets the extract and transmit process option flag. Vaild values are:<br><br>• Extract never, transmit once (N flag)<br>  PutOptions; TO=YES CEUProcessOption; "Unextractable"<br>• Extract once, transmit never (U flag)<br>  PutOptions; EO=YES CEUProcessOption; "Nontransmittable"<br>• Extract repeatedly, transmit never Has no effect<br>• Extract repeatedly, transmit once (R flag)<br>  PutOptions; XMIT=YES CEUProcessOption; "Requestable"<br>• Extract repeatedly, transmit repeatedly (M flag)<br>  PutOptions; MULTXMIT=YES CEUProcessOption; "Multi-transmit"<br><br>In an auto connect scenario, Connect:Enterprise UNIX supplies this flag in the process data within the schedule definition: ACD_INFO/PutOptions. In a remote connect scenario, Connect:Enterprise UNIX can use the Put Option defined in the remote account (RSD). Optional.<br>• If not specified, Connect:Enterprise UNIX uses the Put Option defined in the schedule (ACD_INFO) or in the remote account (RSD).<br>• Default - C (Collected), J (BP protocol flag), Y (binary data format)<br>• If specified, overrides the Connect:Enterprise UNIX Put Option. |
| CEUServerAdapterInstanceName | Normally, this is already defined. |
| CEUTriggerAutoConnect | Indicates that a schedule session should be triggered when the batch is added to the Connect:Enterprise UNIX mailbox. |
| GISDocumentId | If the primary document is not the document to be added, then identify the Sterling Integrator document ID to be added as a Connect:Enterprise UNIX batch. |

## Output from Service to Business Process

The following table describes the information that is placed in process data by the Connect:Enterprise UNIX Server Add service (Message From Service):

| Parameter | Description |
|---|---|
| CEUBatchNumber | The Unique batch identifier within Connect:Enterprise UNIX. In conjunction with the creation date, batches can be uniquely identified within Connect:Enterprise UNIX. |
| CEUBatchDateTime | The creation date of the batch. Format: 'yyyyMMdd'T'hhmm' |
| CEUMailboxId | The Connect:Enterprise UNIX Mailboxes to which the document was added. |
| CEUBatchId | The name to be assigned to the new Connect:Enterprise UNIX batch. |

| Parameter | Description |
|---|---|
| CEUProcessFlagStatus | The process flags as reported by Connect:Enterprise UNIX after the batch is added to the mailbox.C (Collected), R, J (Added by Sterling Integrator), and Y (binary) |
| GISDocumentId | The Sterling Integrator document ID added to the Connect:Enterprise UNIX mailbox. |
| CEUBatchSize | Size of batch in bytes. |
| CEUTriggerAutoconnect | Indicates if a schedule session is triggered when the batch is added to the mailbox. |
| CEUSessionId | The Connect:Enterprise UNIX session ID associated with this activity. |

## Business Process Example

The following example illustrates a small part of the BPML that utilizes the Connect:Enterprise UNIX Server Add service.

```
<!-- Add Msg to Acme mailbox -->
    <operation name="CEU Server Add Service Add Service">
        <participant name="CEUServerAdd"/>
            <output message="AddRequest">
                <assign to="CEUMailboxId">Invoice</assign>
                <assign to="CEUBatchId">MessageName</assign>
                <assign to="CEUTriggerAutoConnect">Yes</assign>
            </output>
            <input message="inmsg">
                <assign to="MessagefromService" from="*"></assign>
            </input>
    </operation>
```

# Chapter 35. Connect:Enterprise UNIX Server Adapter

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

| System name | Connect:Enterprise UNIX Server Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | This adapter does not appear in the GPM because it is not used directly in business processes. |
| Description | This adapter works as a conduit between Connect:Enterprise UNIX and Sterling Integrator. It functions as a business process protocol daemon within Connect:Enterprise UNIX. When enabled, this adapter has a persistent connection to Connect:Enterprise UNIX. It can respond to a schedule notification from Connect:Enterprise UNIX (also called an auto connect) or it can respond to a Sterling Integrator service that initiates an unsolicited session (also called a remote connect). In the case of auto connect, Connect:Enterprise UNIX notifies the adapter when data (which meets the criteria of the schedule definition) is placed in a mailbox. The adapter:<br><br>• Retrieves batch summary data and sets an acknowledgement flag (V flag) in the Connect:Enterprise UNIX system<br><br>• Initiates the business process specified in the Connect:Enterprise UNIX schedule definition<br><br>• Logs the notification event in Connect:Enterprise UNIX<br><br>• Puts the Sterling Integrator business process ID into Connect:Enterprise UNIX. It also adds a hyperlink which can open a new browser window with the Sterling Integrator business process Detail Page.<br><br>In the case of a remote connect, the adapter runs by one or more of the Connect:Enterprise UNIX Server services. |
| Business usage | Use this adapter for all communications with Connect:Enterprise UNIX. Used in conjunction with the Connect:Enterprise UNIX Server services, this adapter can extract or add data from a mailbox, submit queries, change process flags and log messages in Connect:Enterprise UNIX. |
| Usage example | A trading partner adds a batch to a Connect:Enterprise UNIX mailbox that is configured to initiate a Sterling Integrator business process. This adapter receives notification about the added batch and initiates the specified business process. The extract service in the business process runs the adapter to copy the batch from the mailbox and pass the batch into Sterling Integrator for additional processing. |
| Preconfigured? | Yes. One configuration called CEU Server Adapter BP is installed with Sterling Integrator. To use this configuration you must supply a valid Connect:Enterprise UNIX host and port, ID and password, and then you must enable the configuration. |

| Requires third party files? | No |
|---|---|
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• Connect:Enterprise UNIX Server Add Service<br>• Connect:Enterprise UNIX Server Extract Service<br>• Connect:Enterprise UNIX Server Begin Session Service<br>• Connect:Enterprise UNIX Server End Session Service<br>• Connect:Enterprise UNIX Server Batch Status Service<br>• Connect:Enterprise UNIX Server Log Service |
| Application requirements | An active instance of Sterling Integrator and an active instance of Connect:Enterprise UNIX V2.2 or later is required. |
| Initiates business processes? | This adapter initiates a business process when it receives notification from Connect:Enterprise UNIX. The business process is specified in the Connect:Enterprise UNIX schedule definition. |
| Invocation | No |
| Business process context considerations | When initiating a business process due to a Connect:Enterprise UNIX schedule, the business process context contains the AcdName (that is, the Auto Connect Definition name), SessionId and several other important pieces of information. For more information, see *Process Data*. |
| Returned status values | None |
| Restrictions | None |
| Persistence level | None |
| Testing considerations | When the adapter configuration is enabled, it means it is connected to Connect:Enterprise UNIX. Disabled means not connected to Connect:Enterprise UNIX. If Connect:Enterprise UNIX shuts down, the Connect:Enterprise UNIX Server adapter is disabled. However, when Connect:Enterprise UNIX comes back up, the adapter re-enables automatically. For more information, see *Testing the Connect:Enterprise UNIX Adapter*. |

## System Requirements

For a complete list of requirements, see the Connect:Enterprise UNIX *Installation and Administration Guide.*

## Licensing Requirements

You need appropriate adapter licensing in order to use one or more configurations of the Connect:Enterprise UNIX Server adapter.

## How the Connect:Enterprise UNIX Server Adapter Works

The following table shows what tasks are necessary to establish Sterling Integrator and Connect:Enterprise UNIX for interoperability:

| Configuration Task | Application |
|---|---|
| Define an account (RSD–remote site definitions) to allow Sterling Integrator to connect into Connect:Enterprise UNIX.<br>**Note:** This task is optional because the system is delivered with a predefined account called samp_bp. | Connect:Enterprise UNIX |
| Define a schedule (ACD–auto connect definitions) to indicate when to notify Sterling Integrator.<br>**Note:** This task is optional because the system is delivered with a predefined schedule called samp_bp.acd | Connect:Enterprise UNIX |
| Configure the Connect:Enterprise UNIX Server adapter.<br>**Note:** Though the preconfigured adapter, CEU Server Adapter BP, is delivered with the system, you must supply some additional parameters and enable the configuration | Sterling Integrator |
| Test connectivity between Sterling Integrator and Connect:Enterprise UNIX:<br>1. Enable the adapter<br>2. Perform cmusession quick test. | Sterling Integrator and Connect:Enterprise UNIX |
| Run the Interoperability Verification Demonstration. | Sterling Integrator and Connect:Enterprise UNIX |

## Interoperability

The Connect:Enterprise UNIX Server adapter is a Sterling Integrator component that acts as a conduit for connectivity with Connect:Enterprise UNIX and enables interoperability between Sterling Integrator and Connect:Enterprise UNIX.

The following figure shows how Sterling Integrator and Connect:Enterprise UNIX work together to exchange information. It also illustrates load balancing where multiple adapters connect to a single Connect:Enterprise UNIX host and port.
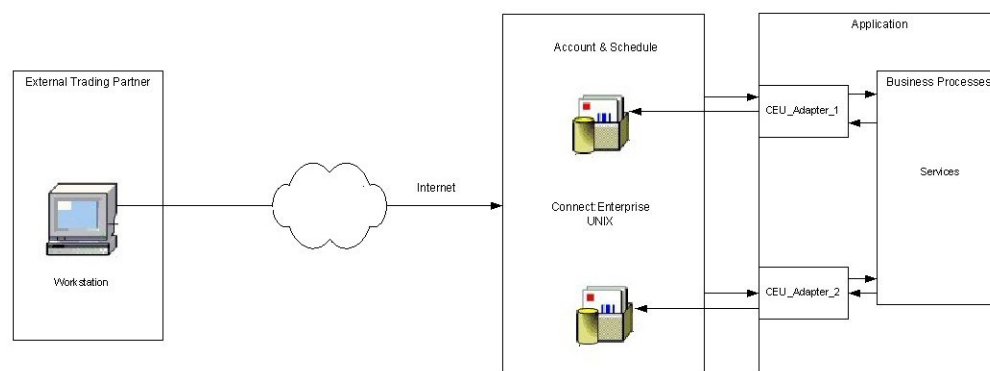


The adapter must be configured with a valid Connect:Enterprise UNIX host, port, ID, and password. After the adapter is configured and enabled, Connect:Enterprise UNIX registers the adapter as a Business Process protocol daemon.

The following describes how the Connect:Enterprise UNIX Server adapter responds to a notification schedule:

1. A configuration of the adapter receives notification from Connect:Enterprise UNIX when data meeting the criteria of a schedule notification is placed in a Connect:Enterprise UNIX mailbox.

2. The adapter retrieves summary data about the batch from the Connect:Enterprise UNIX mailbox according to parameters defined in the schedule.

   For the summary data to be retrieved, the R (requestable) processing flag must be present. However, batch summary data with the following processing flags are not retrieved:

   - P (Transmission in progress)
   - I (Incomplete)
   - D (Deleted)
   - T (Transmitted)
   - V (Sterling Integrator acknowledged)

3. The adapter initiates the Sterling Integrator business process (defined in the Connect:Enterprise UNIX schedule) and places the summary data into process data.

4. The adapter waits for further instructions from the Sterling Integrator business process.

## Process Data

In the case of an auto connect, the batch summary data identifies the adapter that was used, the Connect:Enterprise UNIX SessionId that has been started, and Auto Connect Definition (ACD) information including specific batch summary data. At this point the business document is not in process data. The next step depends upon your business process.

In the case of remote connect, the content of the information placed in process data by the adapter varies depending upon the service that begins the business process.

## CEUServerAdapterInstanceName and SessionId

The configuration name of the Connect:Enterprise UNIX Server adapter (CEUServerAdapterInstanceName) and SessionId are at the top level of the process data for visibility and use by the services in the business process, as in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <CEUServerAdapterInstanceName>BP</CEUServerAdapterInstanceName>
  <SessionId>2057</SessionId>
  <ACD_INFO>
      .
      .
      .
```

These parameters are important because they indicate which Connect:Enterprise UNIX Server adapter configuration started the business process and what session the services will be acting on behalf of. Because you can have multiple configurations of the adapter, each Connect:Enterprise UNIX Server service used in the business process needs to determine which adapter configuration to communicate with so that the business process activity is passed through the correct (requesting) adapter.

**Note:** Because the SessionId is put into process data during a schedule notification, it serves as an implied Begin Session; therefore, you do not need to use a Begin

Session service in a business processes initiated from a schedule notification. However, you must still use an End Session service to bracket all activities within the same SessionId.

**Note:** The CEUServerAdapterInstanceName in process data overrides whatever adapter the services are configured to use. Therefore, if you configure a new Connect:Enterprise UNIX Server adapter you do not need to create a new suite of service configurations to point to the new adapter. The original Connect:Enterprise UNIX Server service configurations are reusable.

## Business Process Parameters Defined in Connect:Enterprise UNIX

The business process parameters in the following example include mailbox information and additional user-defined XML content that is pertinent to the business process. The parameters are entered in Connect:Enterprise UNIX as "Additional Business Process Parameters" when the remote BP (Business Process protocol daemon) block is defined within a schedule.

```
<BusinessProcessParameters format="xml">
  <TP_INFO>
   <Mailboxes>
    <Rejected>bpreject</Rejected>
    <InProcess>bpinproc</InProcess>
    <Invoice>bpinvoic</Invoice>
    <Messages>bpmsg</Messages>
   </Mailboxes>
   <EmailAddress>jane_doe@myco.com</EmailAddress>
  </TP_INFO>
 </BusinessProcessParameters>
```

## Batch Summary Data

Other important pieces of information are the CEUBatchNumber and CEUBatchDateTime. These pieces of information are used when data is extracted. In the following example, you can see the batch summary data for two batches that were named *.po and located in the bporders mailbox. This information is batch summary data only and does not include the actual business document at this point in the business process.

```
.
.
.
<Batches>
   <Batch>
     <CEUBatchNumber>5262</CEUBatchNumber>
     <CEUMailboxId>Process Dataorders</CEUMailboxId>
     <CEUBatchId>CarOrder.po</CEUBatchId>
     <CEUBatchSize>208</CEUBatchSize>
     <CEUBatchDateTime>12/17/03 3:38 PM</CEUBatchDateTime>
     <CEUProcessFlagStatus>C,R,J,Z</CEUProcessFlagStatus>
   </Batch> <Batch>
     <CEUBatchNumber>5263</CEUBatchNumber>
     <CEUMailbox>bporders</CEUMailbox>
     <CEUBatchId>CarOrder.po</CEUBatchId>
     <CEUBatchSize>208</CEUBatchSize>
     <CEUBatchDateTime>12/17/03 3:40 PM</CEUBatchDateTime>
   </Batch>
  </Batches>
 </ACD_INFO>
</ProcessData>
```

## Connect:Enterprise UNIX Server Services

When the Sterling Integrator business process starts, it uses the Connect:Enterprise UNIX Server services to perform activities on batches in or from the Connect:Enterprise UNIX mailbox.

These services, described below, are Sterling Integrator components that work in collaboration with the Connect:Enterprise UNIX Server adapter:

- Connect:Enterprise UNIX Server Add service - Adds a batch to a Connect:Enterprise UNIX mailbox
- Connect:Enterprise UNIX Server Extract service - Extracts a batch from a Connect:Enterprise UNIX mailbox
- Connect:Enterprise UNIX Server Begin Session service - Begins a session with an instance of Connect:Enterprise UNIX
- Connect:Enterprise UNIX Server End Session service - Ends a session with an instance of Connect:Enterprise UNIX
- Connect:Enterprise UNIX Server Batch Status service - Queries and updates batch summary data. This service can reach into the Connect:Enterprise UNIX mailbox to find and even change data including status flags (process and data format), CEUMailboxId and CEUBatchId.
- Connect:Enterprise UNIX Server Log service: Writes log messages for success or error conditions to make them visible in the Connect:Enterprise UNIX interface.

   **Note:** If a business process ran by Connect:Enterprise UNIX is not active, and the Connect:Enterprise UNIX Server End session service was not used, the auto connect session (seen in cmusession) times out after 30 seconds.

## Interoperability Tracking

Activities between Sterling Integrator and Connect:Enterprise UNIX can be tracked from either system.

The Connect:Enterprise UNIX Server adapter writes Sterling Integrator correlation records to enable searches for business processes in Sterling Integrator that contain a particular Connect:Enterprise UNIX AcdName or a SessionId.

The adapter also logs the notification event in Connect:Enterprise UNIX and identifies the Sterling Integrator business process ID that started. This information is visible in the Connect:Enterprise UNIX Detail Account and Schedule reports. The business process ID appears as a hyperlink. Clicking on the business process ID opens a browser window to the Detail page of that business process.

Although a configuration of the Connect:Enterprise UNIX Server adapter is included in Sterling Integrator, you can create a new configuration of the adapter.

Before creating a new configuration of the adapter, review the following tips:

- You can use the Connect:Enterprise UNIX Server adapter named "BP" for your production environment. However, if you plan to use more than one configuration of the adapter with a single installation of Connect:Enterprise UNIX, you must create additional configurations with different names.
- The Connect:Enterprise UNIX schedule that starts a session with Sterling Integrator points to a specific resource. This resource is the registered configuration of the Connect:Enterprise UNIX Server adapter. Therefore, it may be useful to create and name configurations of the adapter according to how

they will be used in Sterling Integrator (for example: SAP, Rosetta, or UCCnet), and then have these multiple adapters connected to a single installation of Connect:Enterprise UNIX.

The following information is needed before you can complete configuration and enable a configuration of the Connect:Enterprise UNIX Server adapter:

- Connect:Enterprise UNIX Host - The host name or IP address of the Connect:Enterprise UNIX instance that the adapter will connect to.
- Connect:Enterprise UNIX Port - The port number the adapter will use to connect to the Connect:Enterprise UNIX application. This should be the main Connect:Enterprise UNIX port.
- Connect:Enterprise UNIX User ID - The user ID required to access Connect:Enterprise UNIX. This is a Connect:Enterprise UNIX remote account.
- Connect:Enterprise UNIX Password - The password required to access Connect:Enterprise UNIX.

## Implementing the Connect:Enterprise UNIX Server Adapter

To implement the Connect:Enterprise UNIX Server adapter, complete the following tasks:

1. Activate your license for the Connect:Enterprise UNIX Server adapter. For information, see *Managing Services and Adapters*.
2. Create a Connect:Enterprise UNIX Server adapter configuration. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Enterprise UNIX Server adapter. For information, see *Configuring the Connect:Enterprise UNIX Server Adapter*.
4. Test the Connect:Enterprise UNIX Server adapter. For information, see *Testing the Connect:Enterprise UNIX Adapter*.
5. Create configurations of any Connect:Enterprise UNIX Server adapter services needed for use with this adapter. See *Related services* in the Overview table for a list of these services and where to find more information about each.
6. Use the Connect:Enterprise UNIX Server adapter and any additional configurations in a business process.

## Configuring the Connect:Enterprise UNIX Server Adapter

To configure the Connect:Enterprise UNIX Server adapter, you must specify settings for the following fields in Sterling Integrator:

| Field | Description |
|---|---|
| Name | Unique and meaningful name for the adapter configuration. Required.<br>**Note:** When enabled, the adapter name registers in Connect:Enterprise UNIX as a Business Process protocol daemon. When you perform a cmusession command in Connect:Enterprise UNIX, this name should be in the list of registered daemons. |
| Description | Meaningful description for the adapter configuration, for reference purposes. Required. |

| Field | Description |
|---|---|
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br><br>• None – You do not want to include this configuration in a group at this time (default)<br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br><br>• 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.<br><br>For more information about service groups see *Managing Services and Adapters*. |
| CEU Host | Host name or IP address of the Connect:Enterprise UNIX installation that the adapter will connect to. |
| CEU Port | Port number the adapter will use to connect to the Connect:Enterprise UNIX application. |
| CEU User ID | User ID required to access the Connect:Enterprise UNIX application when a business process must use a remote connection to Connect:Enterprise UNIX. |
| CEU Password | Password required to access the Connect:Enterprise UNIX application when a business process must use a remote connection to Connect:Enterprise UNIX. |
| Maximum active CEU sessions | Maximum number of sessions that can be active at any given time. The minimum value is 1. |
| Listen Port | Port number on which Sterling Integrator will listen for a Connect:Enterprise UNIX Schedule notification. Valid value is a unique port number on the Sterling Integrator host machine that will not conflict with another process. |

**CAUTION:**
**If the new configuration of the Connect:Enterprise UNIX Server adapter is not enabled, Sterling Integrator cannot communicate with Connect:Enterprise UNIX.**

## Output from Adapter to Business Process

The example below shows what the process data looks like when the Connect:Enterprise UNIX Server adapter runs a business process as a result of a Connect:Enterprise UNIX schedule notification:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
<CEUServerAdapterInstanceName>BP</CEUServerAdapterInstanceName>
  <SessionId>2057</SessionId>
  <ACD_INFO>
    <AcdName>samp_bp.acd</AcdName>
    <SessionId>2057</SessionId>
    <AdapterName>BP</AdapterName>
    <MailboxList>bporders</MailboxList>
    <BatchIdPattern>*.xml</BatchIdPattern>
    <PutOptions>$$CODE=A</PutOptions>
    <BusinessProcessParameters format="xml">
```

```
            <TP_INFO>
              <Mailboxes>
                <Rejected>bpreject</Rejected>
                <InProcess>bpinproc</InProcess>
                <Invoice>bpinvoic</Invoice>
                <Messages>bpmsg</Messages>
              </Mailboxes>
              <EmailAddress>john.doe@myco.com</EmailAddress>
            </TP_INFO>
        </BusinessProcessParameters>
        <Batches>
          <Batch>
            <CEUBatchNumber>9218</CEUBatchNumber>
            <CEUBatchDateTime>1/19/04 9:40 AM</CEUBatchDateTime>
            <CEUBatchSize>208</CEUBatchSize>
            <CEUMailboxId>bporders</CEUMailboxId>
            <CEUBatchId>PO.xml</CEUBatchId>
            <CEUProcessFlagStatus>C,R,J,Y</CEUProcessFlagStatus>
          </Batch>
          <Batch>
            <CEUBatchNumber>9219</CEUBatchNumber>
            <CEUBatchDateTime>1/19/04 9:41 AM</CEUBatchDateTime>
            <CEUBatchSize>208</CEUBatchSize>
            <CEUMailboxId>bporders</CEUMailboxId>
            <CEUBatchId>PO.xml</CEUBatchId>
            <CEUProcessFlagStatus>C,R,J,Y</CEUProcessFlagStatus>
          </Batch>
        </Batches>
      </ACD_INFO>
</ProcessData>
```

## High Concurrency Load Balancing

For scheduled notifications, all communications between Sterling Integrator and Connect:Enterprise UNIX use the adapter configuration in the auto connect request. Throughout the initiated business process, all services use the adapter that began the auto connection.

For remote connection, however, you can create a group of Connect:Enterprise UNIX Server adapters that can be used for load balancing. These adapters should be identically configured except for Name and Listen port. The group name is used for the CEUServerAdapterInstanceName parameter. When a business process uses an adapter from that group for the first service, the same adapter is used to manage all subsequent activity within the same session. At each new session the least recently used adapter in the group is used.

## Testing the Connect:Enterprise UNIX Adapter

If Connect:Enterprise UNIX shuts down while it is connected to Sterling Integrator through the adapter, it will automatically reconnect when Connect:Enterprise UNIX restarts.

To verify that Sterling Integrator and Connect:Enterprise UNIX are communicating:

1. Configure the Connect:Enterprise UNIX Server adapter and save it. If it enables, you are connected.
2. To verify the connection in Connect:Enterprise UNIX, enter one of the following cmusession commands to find a registered configuration of the adapter.

   To list only registered business process protocols:

   cmusession -u <username> -p <password> -g

   To list all registered protocols:

```
cmusession -u <username> -p <password>
```

If the Connect:Enterprise UNIX adapter name does not appear in the list when you perform the cmusession command, the most frequent problems encountered are:

- Invalid host IP address or port number
- The adapter was not enabled when configured

3. If problems continue, turn on the special ceuinterop.log logging from the Operations menu, **Operations** > **System** > **Log**. All services and the adapter log to this file.

In addition, error statements are logged in the appserver log file (jboss.log or weblogic.log).

# Chapter 36. Connect:Enterprise UNIX Server Batch Status Service

The following table provides an overview of the Connect:Enterprise UNIX Server Batch Status service:

| System name | Connect:Enterprise UNIX Server Batch Status Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX |
| Description | This service passes queries and updates to Connect:Enterprise UNIX through the Connect:Enterprise UNIX Server adapter. |
| Business usage | This service is multifunctional. Use it to: <br>• Retrieve summary data for one or more batches. <br>• Determine the state of process status flags and data format status flags for one or more batches. <br>• Change any of the status flags or change a mailbox ID or a batch ID. |
| Usage example | A batch exists in a mailbox but is not extractable by Sterling Integrator because of the process flag settings. For example, the batch is allowed only one extract and it has already been extracted. This service is used to update the process flag on the batch so that it can be extracted again. For more information, see *How the Connect:Enterprise UNIX Server Batch Status Service Works*. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Add service, Extract service, Log service. |
| Application requirements | Active instances of Sterling Integrator and Connect:Enterprise UNIX v2.2 or later are required for this service to work. |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Business process context considerations | The result of a Batch Status service query is a series of data within a STATUS_INFO tag. This service functions as a query only unless specific updates are defined in the service configuration. When this service is used in a business process that is initiated by a schedule notification from Connect:Enterprise UNIX: <br>• It can use the CEUServerAdapterInstanceName and CEUSessionID that is provided automatically in the process data. <br>• A looping business process is normally used to process each element in the auto connect definition ACD_INFO/Batches/Batch. (This information is in the process data.) |

| Returned status values | Returned status values: |
| --- | --- |
| | • Success – Normal execution with the return parameters specified. |
| | • Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. |
| | • Invalid Session – Indicates an invalid session. |
| | • Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. |
| | • Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise Unix Server Adapter. |
| | • CEU Service Error – A generic error associated with the service has occurred. |
| Restrictions | None |
| Persistence level | All |
| Testing considerations | For information, see the business process, CEUInterop_DemoPOProcess. |

## How the Connect:Enterprise UNIX Server Batch Status Service Works

The Connect:Enterprise UNIX Server Batch Status service sends a query to Connect:Enterprise UNIX and, optionally, can be used to update process flags, the CEUBatchId or the CEUMailboxId.

The following example describes what happens when a business process is initiated by a Connect:Enterprise UNIX schedule notification, but then, as a result of a downstream activity's error, the Batch Status service is used to alter the Connect:Enterprise UNIX transmitted (T) flag so that the batch can be extracted again.

1. The Connect:Enterprise UNIX Server adapter initiates a business process in response to a schedule notification from Connect:Enterprise UNIX.

2. The business process runs the extract service which copies the batch from the mailbox and sets the transmitted flag on the batch indicating that Sterling Integrator extracted the batch. In this case the batch is allowed to be transmitted only once.

3. The business process runs a translation of the batch from EDI to XML. However, a map error occurs during translation and the batch is not translated properly.

4. The error triggers the start of the Batch Status service.

5. The Batch Status service passes a message to Connect:Enterprise UNIX to remove the transmitted flag from the batch. The STATUS_INFO returned by Connect:Enterprise UNIX is placed in process data.

Here is an example of output from a Batch Status service that was used to update *all* batches in the bporders mailbox by turning the T flag off. Notice that one batch was returned.

```
<STATUS_INFO>
<QueryParms>
<CEUMailboxId>bporders</CEUBatchNumber>
</QueryParms>
<UpdateParms>
<CEUProcessFlagUpdate>!T</CEUProcessFlagUpdate>
```

```
</UpdateParms>
<Batches>
<Batch>
<CEUBatchNumber>189</CEUBatchNumer>
<CEUBatchDateTime>2/21/04 8:24PM</CEUBatchDateTime>
<CEUBatchSize>123456967</CEUBatchSize>
<CEUMailboxId>bporders</CEUMailboxId>
<CEUBatchId>PriDoc.128M</CEUBatchId>
<CEUProcessFlagStatus>A,R,Z</CEUProcessFlagStatus>
</Batch>
</Batches>
</STATUS_INFO>
```

6. Now the batch can be extracted again.

   The following example illustrates the XPath statements in the subsequent
   Extract service. These statements derive the needed information from the
   STATUS_INFO content in the process data.

```
<!-- Extract batch by xpath to the provided batch number and datetime
          in ProcessData. -->
<operation name="CEU Extract Service">
<participant name="CEUServerExtract"/>
  <output message="ExtractRequest">
    <assign to="CEUBatchNumber"
            from="STATUS_INFO/Batches/Batch/CEUBatchNumber/text()">
    </assign>
    <assign to="CEUBatchDateTime"
            from="STATUS_INFO/Batches/Batch/CEUBatchDateTime/text()">
    </assign>
    <assign to="." from="*"/>
  </output>
  <input message="inmsg">
    <assign to="." from="*"/>
  </input>
</operation>
```

## Implementing the Connect:Enterprise UNIX Server Batch Status Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline
the business tasks to be completed between Connect:Enterprise UNIX and Sterling
Integrator, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Batch Status service, complete
the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this
   service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server Batch
   Status service. For information, see *Configuring the Connect:Enterprise UNIX
   Server Batch Status Service*.
3. Use the Connect:Enterprise UNIX Server Batch Status service in a business
   process.

## Configuring the Connect:Enterprise UNIX Server Batch Status Service

To configure the Connect:Enterprise UNIX Server Batch Status service, you can
optionally specify field settings in Sterling Integrator, but you must specify field
settings in the GPM.

## Sterling Integrator Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Batch Status service:

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

| Field | Description |
|---|---|
| Name | Unique and meaningful name for the service configuration. Required. |
| Description | Meaningful description for the service configuration, for reference purposes. Required. |
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br><br>• None – You do not want to include this configuration in a group at this time (default)<br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br><br>• 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.<br><br>For more information about service groups see *Managing Services and Adapters*. |
| CEU Server Adapter Instance Name(CEUServerAdapterInstanceName) | The instance of the Connect:Enterprise UNIX adapter that the Batch Status service should use. The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP. |

## GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Batch Status service in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| CEUBatchFromDate | Specifies all batches that were created on or after this date. Optional. |
| CEUBatchId | Specifies the name of one batch in Connect:Enterprise UNIX. Optional. |
| CEUBatchNumber | Specifies the batch number. Optional. |
| CEUBatchOriginator | Specifies the originator of the batch. Optional. |
| CEUBatchToDate | Specifies all batches that were created on or before this date. Optional. |
| CEUMailboxId | Specifies the mailbox identification where the batch resides. Optional. |

| Field | Description |
|-------|-------------|
| CEUProcessFlagQuery | Lists the process flags that you are sending the query about. This is a comma-delimited string of process flags. Use an exclamation point (!) before a flag to indicate a lack of the flag. Optional.<br><br>Example: Y, R, T, P, !V would mean to query batches that are binary (Y), requestable (R), transmitted (T) or transmission in progress (P) but which have not been acknowledged by Sterling Integrator (!V). |
| CEUProcessFlagUpdate | Specifies what the update to the flags should be. Needs the same format as in CEUProcessFlagQuery. Optional.<br><br>Example: V, !R, would mean to change the queried batches to acknowledged (!V) and not requestable (!R). |
| CEUServiceAdapterInstanceName | The instance of the Connect:Enterprise UNIX adapter that the Batch Status service should use. This is typically already defined. Required. |
| NewCEUBatchId | Specifies the identification you want to use to update the CEUBatchId. Optional. |
| NewCEUMailboxId | Specifies the identification you want to use to update the CEUMailboxId. Optional. |

**Note:** The service is used as a query only, unless one or more of these three parameters are defined: CEUProcessFlagUpdate, NewCEUBatchId, NewCEUMailboxId.

## Output from Service to Business Process

The following table describes the information placed in process data by the Connect:Enterprise UNIX Server Batch Status service (Message From Service).

| Parameter | Description |
|-----------|-------------|
| CEUBatchNumber | This is the batch number provided to this service as input. (Combined with CEUBatchDateTime this uniquely identifies the batch.) |
| CEUBatchId | This is the name of the batch that was provided to the service as input prior to any update. |
| CEUMailboxId | This is the name of the mailbox that was provided to the service as input prior to any update. |
| CEUProcessFlagQuery | These are the process flags that were queried by the service. |
| CEUProcessFlagUpdate | These are the process flags that were updated by the Batch Status service. |
| CEUBatchDateTime | The batch date and time. Combined with CEUBatchNumber, this uniquely identifies the batch queried or changed by the service. |
| CEUBatchSize | Size of the message in bytes. |
| CEUProcessFlagStatus | This is the status of all process flags as they appear after the Batch Status service. |

## Business Process Example

The following example illustrates a small part of the BPML that utilizes the
Connect:Enterprise UNIX Server Batch Status service.

```
<operation name="CEUServerBatchStatusServiceType">
<participant name="CEUServerBatchStatus"/>
  <output message="CEUServerBatchStatusServiceTypeInputMessage">
      <assign to="CEUBatchNumber" from="ACD_INFO/Batches/Batch[number
            (//numBatches/text())]/CEUBatchNumber/text()">
      </assign>
      <assign to="." from="*">
      </assign>
  </output>
  <input message="inmsg">
      <assign to="." from="*"></assign>
  </input>
</operation>
```

# Chapter 37. Connect:Enterprise UNIX Server Begin Session Service

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

| | |
|---|---|
| System name | Connect:Enterprise UNIX Server Begin Session Service |
| Graphical Process Modeler (GPM) categories | All Services, Application > Sterling Commerce > Connect:Enterprise UNIX |
| Description | This service passes a begin session request to the Connect:Enterprise UNIX Server adapter. This service works with the adapter to initiate a session with Connect:Enterprise UNIX and populate the Sterling Integrator process data with a CEUSessionId. |
| Business usage | Use this service in a business process that initiates an unsolicited session to Connect:Enterprise UNIX. This type of session is called a remote connect. In a remote connect, use this service with the End Session service to group intervening activities in a single Connect:Enterprise UNIX session. |
| Usage example | You want to start a session with Connect:Enterprise UNIX in order to retrieve a group of documents, translate them from EDI to XML and then put them back into Connect:Enterprise UNIX mailboxes. In addition, you want all the activities run by an instance of this business process to be tracked by a single session ID. For more information, see *How the Connect:Enterprise UNIX Server Begin Session Service Works*. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; End Session service, Extract Session service, Add service, Batch Status service, Log service. |
| Application requirements | An active instance of Sterling Integrator and Connect:Enterprise UNIX v2.2 or later are required for this service to work. |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Business process context considerations | This service is not needed when a Connect:Enterprise UNIX schedule initiates the business process. In this case there is an implied Begin Session because the Connect:Enterprise UNIX supplies the CEUSessionId. In a remote connect scenario, if the Begin Session service is not used, each Connect:Enterprise UNIX Server service, such as extract and add, runs with its own CEUSessionId. |

| Returned status values | Returned status values: |
|---|---|
| | • Success – Normal execution with the return parameters specified. |
| | • Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. |
| | • Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. |
| | • Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise Unix Server Adapter. |
| | • CEU Service Error – A generic error associated with the service has occurred. |
| Restrictions | None |
| Persistence level | All |
| Testing considerations | For information, see the predefined business process, CEUInterop_DemoPOAddNoTrigger. |

### How the Connect:Enterprise UNIX Server Begin Session Service Works

The following steps describe what happens when the Connect:Enterprise UNIX Server Begin Session service is used in a business process that initiates a remote connect.

1. The Sterling Integrator business process attempts to initiate a communication session with Connect:Enterprise UNIX through the Connect:Enterprise UNIX Server adapter. The adapter submits a user ID and password to gain access.

2. The adapter returns a CEUSessionId to the Begin Session service. The CEUSessionId is put in the process data.

3. All Connect:Enterprise UNIX Server services (add, extract, batch status and log) that follow the Begin Session service and precede the End Session service are captured under that one CEUSessionId.

### Implementing the Connect:Enterprise UNIX Server Begin Session Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Sterling Integrator, and decide which Connect:Enterprise UNIX Server services are needed.

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

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.

2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server Begin Session service. For information, see *Configuring the Connect:Enterprise UNIX Server Begin Session Service*.

3. Use the Connect:Enterprise UNIX Server Begin Session service in a business process.

## Configuring the Connect:Enterprise UNIX Server Begin Session Service

To configure the Connect:Enterprise UNIX Server Begin Session service, you must specify settings for the following fields in 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 | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br><br>• None – You do not want to include this configuration in a group at this time (default)<br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br><br>• 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.<br><br>For more information about service groups see *Managing Services and Adapters*. |
| CEU Server Adapter Instance Name(CEUServerAdapterInstanceName) | The instance of the Connect:Enterprise UNIX adapter that the Begin Session service should use. The preconfigured Begin Session service uses the preconfigured adapter named CEU Server Adapter BP. |

## Output from Service to Business Process

The following table describes the output from the Connect:Enterprise UNIX Server Begin Session service to the business process:

| Parameter | Description |
|---|---|
| CEUSessionId | Specifies the identifier for the session established between the adapter and the Connect:Enterprise UNIX instance. |

## Business Process Example

The following example illustrates a part of the BPML that uses a Connect:Enterprise UNIX Server Begin Session service to run a remote connect session. All activity is bracketed by the Begin Session and End Session services. The Begin Session service sets CEUSessionId in process data. The Add and Extract services use that CEUSessionId, and End Session removes the CEUSessionId from process data.

```
<process name="CEUInterop_AddExtractService">
  <sequence>
    <operation name="CEUServerBeginSessionServiceType">
    <participant name="CEUServerBeginSession"/>
      <output message="CEUServerBeginSessionServiceTypeInputMessage">
        <assign to="." from="*"></assign
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="CEU Add Service">
    <participant name="TestCEUServerAdd"/>
      <output message="AddRequest">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="CEUAddServiceResults" from="*"></assign>
      </input>
    </operation>
<!-- Remove Primary document to verify extract service -->
    <operation name="Release Service">
    <participant name="ReleaseService"/>
      <output message="ReleaseServiceTypeInputMessage">
        <assign to="TARGET" from="'PrimaryDocument'"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="CEU Extract Service">
    <participant name="TestCEUServerExtract"/>
      <outputmessage="ExtractRequest">
        <assign to="CEUBatchNumber"
            from="CEUAddServiceResults/CEUBatchNumber/text()"></assign>
      </output>
      <input message="inmsg">
        <assign to="CEUExtractServiceResults" from="*"></assign>
      </input>
    </operation>
    <operation name="CEUServerEndSessionServiceType">
    <participant name="CEUServerEndSession"/>
      <output message="CEUServerEndSessionServiceTypeInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

# Chapter 38. Connect:Enterprise UNIX Server End Session Service

The following table provides an overview of the Connect:Enterprise UNIX Server End Session service:

| System name | Connect:Enterprise UNIX Server End Session Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Application > Sterling Commerce > Connect:Enterprise UNIX |
| Description | This service works with the Connect:Enterprise UNIX Server adapter to notify Connect:Enterprise UNIX that the session is over. This service also removes the CEUSessionId from the Sterling Integrator process data. |
| Business usage | Use this service in a business process to end an auto connect session to Connect:Enterprise UNIX. Note that Connect:Enterprise UNIX holds resources for the session until it is ended. In a business process that initiates a remote connect session, use this service in conjunction with the Begin Session service to group multiple activities within one CEUSessionId. |
| Usage example | You want all activities performed in a business process to be tracked, correlated, and reported as a single Connect:Enterprise UNIX session. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, Extract Session service, Add service, Batch Status service, Log service. |
| Application requirements | An active instance of Sterling Integrator and Connect:Enterprise UNIX v2.2 or later are required for this service to work. |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Business process context considerations | Always use this service in business processes initiated by a Connect:Enterprise UNIX schedule notification. Use of this service with the Begin Session service in a business process that initiates a remote connect is optional. For more information, see *How the Connect:Enterprise UNIX Server End Session Service Works*. |

| Returned status values | Returned status values: |
|---|---|
| | • Success – Normal execution with the return parameters specified. |
| | • Invalid Session - Indicates an invalid session. |
| | • Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. |
| | • Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. |
| | • Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise UNIX Server Adapter. |
| | • CEU Service Error – A generic error associated with the service has occurred. |
| Restrictions | None |
| Persistence level | All |
| Testing considerations | For information, see the predefined business process, CEUInterop_DemoPOAddNoTrigger. |

## How the Connect:Enterprise UNIX Server End Session Service Works

In the case of Auto Connect:
- When the business process is initiated by a Connect:Enterprise UNIX schedule notification, place the End Session service after the final Connect:Enterprise UNIX Server service in the business process.
- If the End Session service is not included in the business process, the session with Connect:Enterprise UNIX will end automatically 30 seconds after the business process becomes inactive.
- Non-Connect:Enterprise UNIX activities, such as translations, can be placed before the End Session service. This is useful, for example, if you want the Connect:Enterprise UNIX Server Log service to notify Connect:Enterprise UNIX if errors result from the other activities.

In the case of Remote Connect:
- When the business process initiates a remote connect session with Connect:Enterprise UNIX, you can use the Begin Session and End Session services to group activities under the same CEUSessionId.
- If the Begin/End Session services are not used, each Connect:Enterprise UNIX Server service, such as Add or Extract, runs with its own CEUSessionId. Each activity is represented in Connect:Enterprise UNIX reporting as separate, unrelated entries.

## Implementing the Connect:Enterprise UNIX Server End Session Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Sterling Integrator, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server End Session service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server End Session service. For information, see *Configuring the Connect:Enterprise UNIX Server End Session Service*.
3. Use the Connect:Enterprise UNIX Server End Session service in a business process.

## Configuring the Connect:Enterprise UNIX Server End Session Service

To configure the Connect:Enterprise UNIX Server End Session service, you must specify settings for the following fields in 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 | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are: <br>• None – You do not want to include this configuration in a group at this time (default)<br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br>• 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.<br>For more information about service groups see *Managing Services and Adapters*. |
| CEU Server Adapter Instance Name(CEUServerAdapterInstanceName) | The instance of the Connect:Enterprise UNIX adapter that the End Session service should use. The preconfigured End Session service uses the preconfigured adapter named CEU Server Adapter BP. |

## Output from Business Process to Service

The following table describes the output from the business process to the Connect:Enterprise UNIX Server End Session service:

| Parameter | Description |
|---|---|
| CEUSessionId | Specifies the identifier for the session established between the adapter and the Connect:Enterprise UNIX instance. |

## Business Process Example

A Business Process protocol daemon that runs by a schedule notification and runs synchronously where all activity is under the CEUSessionId that is found in process data:

```
<process name="CEUInterop_ACDExtractLoop">
  <rule name="notFinished">
    <condition>number(counter/text()) &gt; 0</condition>
  </rule>
<sequence name="Sequence Start">
<assign to="counter" from="count(//ACD_INFO/Batches/Batch/BatchNumber/text())">
    </assign>
<sequence name="theLoop">
    <!-- Remove Primary document to verify extract service -->
    <operation name="Release Service">
    <participant name="ReleaseService"/>
      <output message="ReleaseServiceTypeInputMessage">
        <assign to="TARGET" from="'PrimaryDocument'"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
<operation name="CEU Extract Service">
    <participant name="TestCEUServerExtract"/>
      <output message="ExtractRequest">
        <assign to="CEUBatchNumber" from="ACD_INFO/Batches/
            Batch[number(//counter/text())]/BatchNumber/text()">
        </assign>
      </output>
      <input message="inmsg">
        <assign to="MessageFromExtractService" from="*" append="true"/>
        <assign to="." from="PrimaryDocument" append="true"/>
      </input>
    </operation>
<assign to="counter" from="number(counter/text()) - 1"></assign>
    <choice>
      <select>
        <case ref="notFinished" activity="doRepeat"/>
      </select>
<repeat name="doRepeat" ref="theLoop"/>
    </choice>
  </sequence>
<operation name="CEUServerEndSessionServiceType">
    <participant name="CEUServerEndSession"/>
      <output message="CEUServerEndSessionServiceTypeInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

# Chapter 39. Connect:Enterprise UNIX Server Extract Service

The following table provides an overview of the Connect:Enterprise UNIX Server Extract service:

| System name | Connect:Enterprise UNIX Server Extract Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX |
| Description | This service passes a batch extract request to the Connect:Enterprise UNIX Server adapter. This service works with the adapter to copy data from a Connect:Enterprise UNIX mailbox to the Sterling Integrator. The batch becomes the primary document in the business process. |
| Business usage | Use this service to take business documents out of a Connect:Enterprise UNIX mailbox and copy them into Sterling Integrator for processing. |
| Usage example | You want all claims that are added to your mailbox to be translated from EDI into XML and then routed to the appropriate claims processing department. For more information, see *How the Connect:Enterprise UNIX Server Extract Service Works*. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Add service, Batch Status service, Log service. |
| Application requirements | An active instance of Sterling Integrator and Connect:Enterprise UNIX v2.2 or later are required for this service to work. |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |

| | |
|---|---|
| Business process context considerations | This service can be used in at least two different situations.<br><br>• Where the business process is initiated by a schedule notification from Connect:Enterprise UNIX, also called an auto-connect.<br><br>In this case, a looping business process is normally used to process each element in the auto-connect definition ACD_INFO/Batches/Batch. (This information is in the process data.)<br><br>This service can also use XPath statements to find the CEUBatchNumber and CEUBatchDateTime in the process data.<br><br>Also in this case, the process data automatically includes the CEUServerAdapterInstanceName and a CEUSessionId so that a Connect:Enterprise UNIX Server Begin Session service is not needed.<br><br>• Where the business process makes an unsolicited request to Connect:Enterprise UNIX, also called a remote connection.<br><br>In this case this service should be preceded by the Begin Session service. It could also be preceded by a Batch Status service to acquire the necessary CEUBatchNumber and CEUBatchDateTime. |
| Returned status values | Returned status values:<br><br>• Success – Normal execution with the return parameters specified.<br><br>• Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value.<br><br>• Invalid Session – Indicates an invalid session.<br><br>• Missing Mandatory Parameter Error – A mandatory parameter for the service is missing.<br><br>• Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise UNIX Server Adapter.<br><br>• CEU Service Error – A generic error associated with the service has occurred. |
| Restrictions | None |
| Persistence level | All |
| Testing considerations | For information, see the business process, CEUInterop_DemoPOProcess. |

## How the Connect:Enterprise UNIX Server Extract Service Works

The following describes what happens when this service is used in a business process that is initiated by a Connect:Enterprise UNIX schedule notification.

1. The Connect:Enterprise UNIX Server adapter initiates a business process in response to a schedule notification from Connect:Enterprise UNIX.

2. The business process specifies to extract data from a Connect:Enterprise UNIX mailbox using the Connect:Enterprise UNIX Server Extract service. This service works through the Connect:Enterprise UNIX Server adapter.

3. Using the information from the schedule notification, the Connect:Enterprise UNIX Server adapter extracts the batch that matches the CEUBatchNumber and CEUBatchDateTime in process data.

4. Connect:Enterprise UNIX sets the Transmitted (T) flag on the batch indicating that Sterling Integrator extracted the batch.

5. Sterling Integrator adds a log entry into Connect:Enterprise UNIX that includes the business process ID of this business process.

6. The service adds the batch into the business process as a primary document.

7. Other batch details are added to process data.

   For information, see *Output from Service to Business Process*.

8. Sterling Integrator sets the correlations between the business process that initiated the request and batch summary data from Connect:Enterprise UNIX. These include: CEUBatchId, CEUMailboxId, CEUBatchNumber, schedule definition name and CEUSessionId. These correlations allow subsequent searches for this information using the Central Search function.

9. The system performs additional activities as indicated within the business process. For example, the contents of the batch could be translated into another format.

## Implementing the Connect:Enterprise UNIX Server Extract Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Sterling Integrator, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Extract service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.

2. Configure the pre-installed copy of the Connect:Enterprise UNIX Server Extract service. For information, see *Configuring the Connect:Enterprise UNIX Server Extract Service*.

3. Use the Connect:Enterprise UNIX Server Extract service in a business process.

## Configuring the Connect:Enterprise UNIX Server Extract Service

To configure the Connect:Enterprise UNIX Server Extract service, you must specify field settings in Sterling Integrator and in the GPM.

### Sterling Integrator Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Extract service:

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

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

| Field | Description |
|---|---|
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br><br>• None – You do not want to include this configuration in a group at this time (default)<br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br><br>• 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.<br><br>For more information about service groups see *Managing Services and Adapters*. |
| CEU Server Adapter Instance Name(CEUServerAdapterInstanceName) | The instance of the Connect:Enterprise UNIX adapter that the Extract service should use. The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP. |
| Document Storage(DatabaseOrFileSystem) | Defines how the document will be stored in the system. Required. Valid values are:<br><br>• System Default<br><br>• Database<br><br>• File System<br><br>Choose file system when handling very large batches. |

## GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Extract service in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| CEUBatchNumber | Numeric identifier for a batch within Connect:Enterprise UNIX |
| CEUBatchDateTime | Combined with CEUBatchNumber, this provides unique access to a batch. That is, these two parameters identify what batch is to be extracted. |
| CEUServerAdapterInstanceName | Normally, this is already defined. |
| DatabaseOrFileSystem | Normally, this is already defined. |

The required parameters CEUBatchNumber and CEUBatchDateTime can be obtained using XPath expressions when the business process is initiated by an auto connect. The following are examples of such XPath expressions:

```
<assign to="CEUBatchNumber" from="ACD_INFO/Batches/Batch[number
        (//loop_counter/text())]/CEUBatchNumber/text()"></assign>
<assign to="CEUBatchDateTime" from="ACD_INFO/Batches/Batch
        [number(//loop_counter/text())]/CEUBatchDateTime/text()"></assign>
```

## Output from Business Process to Service

The following table describes the output from the business process to the Connect:Enterprise UNIX Server Extract service (Message To Service):

| Parameter | Description |
| --- | --- |
| CEUBatchNumber | Numeric identifier for a batch within Connect:Enterprise UNIX |
| CEUBatchDateTime | Combined with CEUBatchNumber provides unique access to a batch |

## Output from Service to Business Process

The following table describes the information that is placed in process data by the Connect:Enterprise UNIX Server Extract service (Message From Service):

| Parameter | Description |
| --- | --- |
| PrimaryDocument | Batch payload |
| CEUBatchNumber | Echoes the batch number provided to this service as input. |
| CEUBatchDateTime | Echoes the batch date and time provided to this service as input. |
| CEUBatchSize | Size of the message in bytes. |
| CEUBatchId | Identifier (name) of the batch in Connect:Enterprise UNIX. |
| CEUMailboxId | The Connect:Enterprise UNIX Mailbox ID from which the batch was extracted. |
| CEUSessionId | Connect:Enterprise UNIX session that the extract activity is associated with. |
| CEUProcessFlagStatus | Depicts the flag settings for the batch before the extract occurred. Comma delimited characters. For more information, see your Connect:Enterprise UNIX documentation. |

## Business Process Example

The following example illustrates a small part of the BPML that utilizes the Connect:Enterprise UNIX Server Extract service:

```
<operation name="CEU Extract Service">
      <participant name="CEUServerExtract"/>
        <output message="ExtractRequest">
          <assign to="CEUBatchNumber" from="ACD_INFO/Batches/Batch[
                   number(//loop_counter/text())]/CEUBatchNumber/text()">
</assign>
          <assign to="CEUBatchDateTime" from="ACD_INFO/Batches/Batch[
                   number(//loop_counter/text())]/CEUBatchDateTime/text()">
</assign>
          <assign to="." from="*"/>
         <!-- <assign to="DatabaseOrFileSystem">db</assign> -->
        </output>
        <input message="inmsg">
          <assign to="CEUExtractServiceResults" from="*" append="true"/>
          <assign to="." from="PrimaryDocument"/>
        </input>
      </operation>
```

# Chapter 40. Connect:Enterprise UNIX Server Log Service

The following table provides an overview of the Connect:Enterprise UNIX Server Log service:

| | |
|---|---|
| System name | Connect:Enterprise UNIX Server Log Service |
| Graphical Process Modeler (GPM) categories | All Services, Applications > Sterling Commerce > Connect:Enterprise UNIX |
| Description | This service works with the Connect:Enterprise UNIX Server adapter to write success and error messages to Connect:Enterprise UNIX so they are visible in the Connect:Enterprise UNIX interface for tracking. |
| Business usage | You use this service in a business process to provide information to Connect:Enterprise UNIX in the event of an error condition that results from the execution of the business process. It can also be used to log success messages. |
| Usage example | You want the Connect:Enterprise UNIX administrator to be notified when downstream processing of some batch fails or succeeds. For more information, see *How the Connect:Enterprise UNIX Server Log Service Works*. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Connect:Enterprise UNIX Server adapter and the following Connect:Enterprise UNIX Server services; Begin Session service, End Session service, Add service, Extract service, Batch Status service. |
| Application requirements | An active instance of Sterling Integrator and Connect:Enterprise UNIX v2.2 or later are required for this service to work. |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Business process context considerations | When the business process is initiated by a schedule notification from Connect:Enterprise UNIX, also called an auto connect, this service should use the CEUServerAdapterInstanceName and CEUSessionId provided automatically in process data. Normally, a looping business process is used to process each element in the auto connect definition ACD_INFO/Batches/ Batch. (This information is in the process data.) |

| Returned status values | Returned status values: |
|---|---|
| | • Success – Normal execution with the return parameters specified. |
| | • Invalid Parameter Error – An error has been made in the passing of parameters to this service, possibly a malformed date or numeric value. |
| | • Invalid Session – Indicates an invalid session. |
| | • Missing Mandatory Parameter Error – A mandatory parameter for the service is missing. |
| | • Adapter Not Found – An error indicating that the service was unable to locate the Connect Enterprise UNIX Server Adapter. |
| | • CEU Service Error – A generic error associated with the service has occurred. |
| Restrictions | None |
| Persistence level | All |
| Testing considerations | For information, see the business process, CEUInterop_DemoPOProcess. |

## How the Connect:Enterprise UNIX Server Log Service Works

The following describes an example of what happens when this service is used in a business process to log success and error messages to the Connect:Enterprise UNIX interface.

1. A business process extracts a batch from Connect:Enterprise UNIX and passes it along the business process for EDI-to-XML translation.
2. An error occurs during translation and the batch is not translated properly.
3. As a result of the error, the Connect:Enterprise UNIX Server Log service writes a log message about the translation failure to Connect:Enterprise UNIX.
4. When the Connect:Enterprise UNIX administrator views information about the activity, the message provided by the Log service is displayed with a hyperlink to Sterling Integrator.

## Implementing the Connect:Enterprise UNIX Server Log Service

Before implementing any specific Connect:Enterprise UNIX Server services, outline the business tasks to be completed between Connect:Enterprise UNIX and Sterling Integrator, and decide which Connect:Enterprise UNIX Server services are needed.

To implement the Connect:Enterprise UNIX Server Log service, complete the following tasks:

1. Configure the Connect:Enterprise UNIX Server adapter to be used with this service. For information, see *Connect:Enterprise UNIX Server Adapter*.
2. You can use the pre-installed copy of the Connect:Enterprise UNIX Server Log service, or create a new copy. You might want to create a new copy if you are using more than one copy of the Connect:Enterprise UNIX Server adapter, for example. For information, see *Managing Services and Adapters*.
3. Configure the Connect:Enterprise UNIX Server Log service. For information, see *Configuring the Connect:Enterprise UNIX Server Log Service*.
4. Use the Connect:Enterprise UNIX Server Log service in a business process.

## Configuring the Connect:Enterprise UNIX Server Log Service

To configure the Connect:Enterprise UNIX Server Log service, you must specify field settings in Sterling Integrator and in the GPM. This is only necessary if you create a new copy of the service; the pre-installed copy of the service does not require any configuration.

### Sterling Integrator Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Log service:

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

| Field | Description |
|---|---|
| Name | Unique and meaningful name for the service configuration. Required. |
| Description | Meaningful description for the service configuration, for reference purposes. Required. |
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are: <br><br>• None – You do not want to include this configuration in a group at this time (default) <br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration. <br><br>• 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. <br><br>For more information about service groups see *Managing Services and Adapters*. |
| CEU Server Adapter Instance Name(CEUServerAdapterInstanceName) | The instance of the Connect:Enterprise UNIX adapter that the Log service should use. The preconfigured service uses the preconfigured adapter named CEU Server Adapter BP. |

### GPM Configuration

The following table describes the fields used to configure the Connect:Enterprise UNIX Server Log service in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| CEUBatchId | Name of the batch to which the error or success message relates. Viewed in the Connect:Enterprise UNIX tracking reports as informational. Populates columns in Connect:Enterprise UNIX detailed reporting screens on the administration user interface. Optional |

| Field | Description |
|---|---|
| CEUBatchNumber | This is provided to the service as input. Viewed in the Connect:Enterprise UNIX tracking reports as informational. Populates columns in Connect:Enterprise UNIX detailed reporting screens on the administration user interface. Optional |
| CEUMailboxId | Connect:Enterprise UNIX mailbox where the batch resides. Viewed in the Connect:Enterprise UNIX tracking reports as informational. Populates columns in Connect:Enterprise UNIX detailed reporting screens on the administration user interface. Optional |
| CEULogStatusCode | The status code that can be viewed in the Connect:Enterprise UNIX tracking statistics column. Success (0) and Error (FA). Required |
| CEUServerAdapterInstanceName | Normally, this is already defined. |

## Output from Business Process to Service

The previous table describes the output from the business process to the Connect:Enterprise UNIX Server Log service (Message To Service).

## Business Process Example

The following example illustrates a small part of the BPML that utilizes the Connect:Enterprise UNIX Server Log service.

```
<operation name="CEU Log Service">
   <participant name="CEUServerLog"/>
     <output message="LogRequest">
        <assign to="CEUMailboxId" from="ACD_INFO/Batches/Batch[number
         (//loop_counter/text())]/CEUMailboxId/text()"></assign>
        <assign to="CEUBatchId" from="ACD_INFO/Batches/Batch[number
         (//loop_counter/text())]/CEUBatchId/text()"></assign>
        <assign to="CEUBatchNumber" from="ACD_INFO/Batches/Batch[number
         (//loop_counter/text())]/CEUBatchNumber/text()"></assign>
        <assign to="CEULogStatusCode">Error</assign>
        <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
        <assign to="." from="*"></assign>
     </input>
</operation>
```

# Chapter 41. Check Expire Service

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

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

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

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

## How the Check Expire Service Works

The Check Expire service runs on a schedule (by default daily at 2:30 a.m.) and looks at each checked in certificate to compare its expiration date against the value set in Sterling Integrator (the default is 14 days). Any certificates that have already expired or will expire sometime within the specified number of days are listed on

the Admin Console Home page in System Alerts, System Alerts on Dashboard, and the Event Viewer under the Operator menu. The alerts remain in the system for 60 days.

## Implementing the Check Expire Service

There is no implementation necessary for the Check Expire service, unless you want to change the number of days prior to expirations that you receive alerts, or the schedule. To do so, edit the service configuration called Check Expire Service. The following table describes the fields used to configure the Check Expire adapter in 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 a Service Group to associate with this adapter. Valid values:<br>• None – You do not want to include this configuration in a group at this time. Default.<br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br>• 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.<br>• For more information about service groups, see *Managing Services and Adapters*. |
| Run as User | Type the user ID to associate with the schedule, or select a user ID from the list. Valid value is any valid Sterling Integrator user ID. |
| Use 24 Hour Clock Display | If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock. |

| Field | Description |
|---|---|
| Schedule | By default, the Check Expire service and business process are scheduled to run every day at 2:30 a.m. Valid values: <br><br>• Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule. <br><br>• Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. <br><br>• Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. <br><br>• Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. <br><br>• Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |
| Expire Days | Number of days before a certificate expires that you will begin receiving alerts. Default is 14. Required. |
| Output Method | Type of alert to use when a certificate is about to expire. Valid values are Email, EVENT and User Message. |
| Output Email Address (Output Method = Email) | E-mail address to use when the alert output method is Email. |
| User(s) Settings (Output Method = User Message) | Users for whom an alert message will be created when a certificate is about to expire. Used when the alert output method is User Message. |
| Exclusion Settings | Certificates to exclude from reporting. |

## Output from Service to Business Process

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

| CHECKEXPIRE | Specifies the type of alert to use when a certificate is about to expire. Valid values are Email, EVENT and User Message. |
|---|---|

| EXPIRE_DAYS | Specifies when you start receiving alerts that a certificate is about to expire. For example, if this parameter is set to 7 days, you will start receiving alerts seven days before the certificate is scheduled to expire. Default is 14. |
|---|---|
| OUTPUT_MAIL | Specifies the e-mail address to use when using an e-mail alert. |

## Business Process Example

This is the Schedule_CheckExpireService business process as installed with Sterling Integrator:

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

# Chapter 42. Command Line Adapter

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

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

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

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

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

## How the Command Line Adapter Works

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

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

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

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

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

5. Sterling Integrator performs the next operation in the business process.

## Implementing the Command Line Adapter

You can implement a Command Line adapter to do the following:
- Execute commands using the command line from within a business process.
- Invoke the Command Line adapter on a schedule and then start a new business process using the output from the adapter.

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

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

## Before You Begin

Before you begin to implement the Command Line adapter, complete the following tasks:
1. Create and test the command line program or command to make sure that it works.
2. Determine the working directory where you will be processing your commands.

## Process Overview

To implement the Command Line adapter, complete the following tasks:
1. Create a Command Line adapter configuration. See *Managing Services and Adapters*.
2. Configure the Command Line adapter. See *Configuring the Command Line Adapter*.
3. Create and enable a business process that includes the Command Line adapter.

   **Note:** If you are configuring a Command Line adapter to start a business process, create the business process before configuring the adapter.
4. Test the business process and the adapter.
5. Run the business process.

## Configuring the Command Line Adapter

To create a Command Line adapter configuration, you must specify field settings in Sterling Integrator and in the 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 Command Line 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. Some fields can be configured in the GPM, if not selected here. Regardless of where they are configured, they can be overridden using BPML.

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

| Field | Description |
|---|---|
| Command Line (cmdLine) | Command line process you want to run. Do one of the following: |
| | • If you want to set this parameter in the GPM/business process, leave the field blank. |
| | • Type the command line process in this field exactly as you would from the command line. |
| | • If you want to use a command that redirects input or output (through the use of >, <, or \|), you must do so using a script file. |
| | • If you do not know the input or output file name, type the following parameters in the command line process as placeholders: |
| |   • `$Input` |
| |   • `$Output` |
| | These parameters are typed directly in the command line process. You can use these parameters on the command line in any order and multiple times if necessary. At run time, they are replaced with the actual file name. |
| | • If you want to enter user parameters, use the following placeholders: $0 – $9. These placeholders are resolved by the parm0 – parm9 parameters defined in the GPM or using BPML. |
| | **Note:** If $Input or $Output resolves to a filename that contains one or more spaces, automatic quoting will be performed before the command line is executed. For example, If the original command line was `test.sh $Input`, and $Input resolves to file 1, then the final command line, before execution, will be `test.sh "file 1"`. Therefore, do not put quotes around $Input or $Output. |
| | **Note:** An example of a command line entry is *test.sh $Input $Output $0 $1 $2 $3 $4 $5 $6 $7 $8 $9.* This runs the shell script `test.sh` taking an input file, using ten parameters, and producing an output file. |
| Working Directory (workingDir) | Location of the directory to use for executing the command line process. Optional. Default is the current working directory of the JVM running CLA2Client.jar. |
| | **CAUTION:** |
| | **Using this adapter to call a Unix script modifies the directory path of the environment variable LD_LIBRARY_PATH. To keep your current path, your script should include either the LD_LIBRARY_PATH path or a reference to your .profile (which includes the LD_LIBRARY_PATH path).** |
| | **Note:** The CLAClient.jar is no longer used. |

| Field | Description |
|---|---|
| Turn on debugging messages? (cla2_debug) | Turn on debugging for this adapter instance? Valid values:<br><br>• Yes (true) – Logging is turned on and the messages are written to the system log.<br>• No (false) – Default.<br><br>**Note:** This turns on debugging for this specific adapter instance. These messages are logged in the system log in the *install_dir*/logs directory. This parameter is read-only in the GPM. |
| Wait on the process to complete before continuing? (waitOnProcess) | Wait on the process to complete before continuing the business process. Valid values:<br><br>• Yes (true) – If the value is Yes, a status report is created if any stdout/stderr is generated by the process. If an error occurs while the service is processing output data, the advanced status contains the error message instead of the return code value.<br>• No (false)<br><br>**Note:** If **Use the output generated by the command line process** is set to Yes, the value of this parameter is assumed to be Yes because the service cannot use output if it does not wait for the process to complete. This parameter is read-only in the GPM. |
| Does this service start a business process? (bootstrap) | Whether the service starts a business process. Required. Valid values:<br><br>• Yes (true)<br>• No (false)<br><br>**Note:** This parameter is read-only in the GPM. |
| Business process (initialWorkFlowName) | Business process you want the Command Line adapter to start. This field is required only if you selected Yes in *Does this service start a business process?* . If you prefer to configure this parameter in the GPM, select Not Applicable. |
| Document Storage Type (docStorageType) | Defines how the document is stored in the system. Required when the adapter starts a business process. Valid values:<br><br>• System Default – Default<br>• Database<br>• File System<br><br>**Note:** For more information on document storage types, see *Managing Services and Adapters*. |

| Field | Description |
|---|---|
| Run as User | 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. <br>**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. |
| 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 business process invoked by the Command Line adapter. The Schedule field only displays as an option if *Does this service start a business process?* is set to Yes. Valid values: <br><br>• Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule. <br><br>• Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. <br><br>• Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. <br><br>• Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. <br><br>• Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |

| Field | Description |
|---|---|
| Does the command line process require an input file? (useInput) | Defines whether the command line process requires an input file? Required. Valid values:<br><br>• Yes (true) – The primary document of the current business process context is written out to the file system in the working directory and is used as input to the process. Default.<br><br>• No (false) – No file is written to disk even if a document exists in the business process context.<br><br>**Note:** This parameter is read-only in the GPM. |
| Input File Name (inputName) | Input file name, if the command line process requires an input file. Any occurrences of $Input in the command line are replaced with this name. Optional. If you leave this field blank, the default is the primary document name.<br>**Note:** It is important to have a unique input file name for all concurrently running instances of Command Line adapters. If more than one instance of the Command Line adapter can be executing at the same time, you must create a dynamic, unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine. |
| Delete input file after process completes? (inputDelete) | Defines whether the input file is deleted after the process completes? Valid values:<br><br>• Yes (true) – Default<br><br>• No (false)<br><br>**Note:** To delete the input file, **Wait on the process to complete before continuing?** must also be Yes. This parameter is read-only in the GPM. |
| Use the output generated by the command line process? (useOutput) | Use output generated by the command line process? Required. Valid values:<br><br>• Yes (true) – The adapter will attempt to read the output of the process. If bootstrapping a workflow, the file will become the primary document in the new workflow. If not bootstrapping, the file will become the primary document of the current workflow. Default .<br><br>• No (false) – No file is read into the business process context even if one is generated by the command line process.<br><br>**Note:** This parameter is read-only in the GPM. |

| Field | Description |
|---|---|
| Output File Name (outputName) | Output file name, if you want to use the output generated by the command line process. Any occurrences of $Output in the command line are replaced with this name. Optional. If you leave this field blank, the default is the business process primary document name. **Note:** It is important to have a unique output file name for all concurrently running instances of command line adapters. If more than one instance of the Command Line adapter can be executing at the same time, you must create a dynamic unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine. |
| Delete output file after process completes? (outputDelete) | Specifies whether the output file is deleted after it is collected?Valid values:<br><br>• Yes (true) – Default<br><br>• No (false)<br><br>**Note:** This parameter is read-only in the GPM. |

## GPM Configuration

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

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

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

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

| Field | Description |
|---|---|
| Config (participant name) | Name of the adapter configuration. Required. |
| env0 | An environment variable in the form name=value. Optional. Any value is valid. |
| env1 | An environment variable in the form name=value. Optional. Any value is valid. |
| env2 | An environment variable in the form name=value. Optional. Any value is valid. |
| env3 | An environment variable in the form name=value. Optional. Any value is valid. |
| env4 | An environment variable in the form name=value. Optional. Any value is valid. |
| env5 | An environment variable in the form name=value. Optional. Any value is valid. |
| env6 | An environment variable in the form name=value. Optional. Any value is valid. |
| env7 | An environment variable in the form name=value. Optional. Any value is valid. |
| env8 | An environment variable in the form name=value. Optional. Any value is valid. |
| env9 | An environment variable in the form name=value. Optional. Any value is valid. |
| keepPath | Normally, any path information is stripped off the filename to allow for platform independence. This parameter allows you to keep the entire path. Optional. Valid values:<br>• Yes – Path information is retained<br>• No – Path information is stripped off |
| parm0 | Resolves the $0 placeholder. Optional. Any value is valid. |
| parm1 | Resolves the $1 placeholder. Optional. Any value is valid. |
| parm2 | Resolves the $2 placeholder. Optional. Any value is valid. |
| parm3 | Resolves the $3 placeholder. Optional. Any value is valid. |
| parm4 | Resolves the $4 placeholder. Optional. Any value is valid. |
| parm5 | Resolves the $5 placeholder. Optional. Any value is valid. |
| parm6 | Resolves the $6 placeholder. Optional. Any value is valid. |
| parm7 | Resolves the $7 placeholder. Optional. Any value is valid. |
| parm8 | Resolves the $8 placeholder. Optional. Any value is valid. |
| parm9 | Resolves the $9 placeholder. Optional. Any value is valid. |

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

### Output from Adapter to Business Process

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

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

## Usage Examples

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

## Invoking the Command Line Adapter to Run a Shell Script

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

- When this example configuration is used, a shell script called "test.sh" (which resides in the /home directory) is run.
- The program requires the input filename as the first parameter, the output filename as the second parameter, and three other parameters.
- Because the useInput variable is set to true and the inputName variable is blank, the name of the primary document replaces the $Input placeholder.
- Because the useOutput variable is set to true and the outputName variable is blank, the $Output placeholder is replaced with the name of the primary document.
- If the document name in the workflow context is "data.txt" in this example, the command line becomes /home/test.sh data.txt data.txt VAR1 USER 10 at run-time.
- The name of the primary document is passed as the input file to the shell script program on the command line.
- The name of the primary document is passed as the output file to the shell script program on the command line.

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

## GPM Example

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



## Business Process Modeling Language (BPML) Example

The following example illustrates the same business process using BPML.

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

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

## Invoking the Command Line Adapter

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

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

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

   **Note:** The CLA2Client.jar does not need to be moved if the remote version is on the same machine.
3. Start the remote adapter, enter:

   `java –jar CLA2Client.jar <port>`

   Example: `java –jar CLA2Client.jar 15699` The *port* can be any port that is not in use. This is the value to assign in the Remote Port field in the service configuration. See *Remote Port (remotePort)*. If you upgrade Sterling Integrator at any time, you will also need to use the corresponding new CLA2Client.jar file to avoid a "ClassConflict" error.

   **Note:** You may also run CLA2Client.jar with these commands:
   * (UNIX or iSeries) `./startCmdLine2.sh`
   * (Windows service) `startCLA2WindowsService.cmd`

## Stopping the Command Line Adapter

If Sterling Integrator is shut down with (Windows) stopWindowsService.cmd or (UNIX and iSeries) hardstop.sh script, the Command Line adapter also shuts down.

You can also stop the Command Line Adapter 2 with these commands:
* (UNIX or iSeries) `./stopCmdLine2.sh`
* (Windows service) `stopCLA2WindowsService.cmd`

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

# Chapter 43. Command Line Adapter 2

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

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

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

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

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

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

## How the Command Line Adapter 2 Works

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

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

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

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

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

5. Sterling Integrator performs the next operation in the business process.

## Implementing the Command Line Adapter 2

You can implement a Command Line Adapter 2 to do the following:
- Execute commands using the command line from within a business process.
- Invoke the Command Line Adapter 2 on a schedule and then start a new business process using the output from the adapter.

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

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

## Before You Begin

Before you begin to implement the Command Line Adapter 2:
1. Create and test the command line program or command to make sure that it works.
2. Determine the working directory where you will be processing your commands.

## Process Overview

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

   **Note:** If you are configuring a Command Line Adapter 2 to start a business process, create the business process before configuring the adapter.
4. Test the business process and the adapter.
5. Run the business process.

## Configuring the Command Line Adapter 2

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

**The Application Configuration**

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

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

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

| Field | Description |
|---|---|
| Command Line (cmdLine) | Command line process you want to run. Do one of the following:<br><br>• If you want to set this parameter in the GPM/business process, leave the field blank.<br><br>• Type the command line process in this field exactly as you would from the command line.<br><br>• If you want to use a command that redirects input or output (through the use of >, <, or \|), you must do so using a script file.<br><br>• If you do not know the input or output file name, type the following parameters in the command line process as placeholders:<br><br>  • `$Input`<br><br>  • `$Output`<br><br>  These parameters are typed directly in the command line process. You can use these parameters on the command line in any order and multiple times if necessary. At run time, they are replaced with the actual file name.<br><br>• If you want to enter user parameters, use the following placeholders: $0 – $9. These placeholders are resolved by the parm0 – parm9 parameters defined in the GPM or using BPML.<br><br>**Note:** If $Input or $Output resolves to a filename that contains one or more spaces, automatic quoting will be performed before the command line is executed. For example, If the original command line was `test.sh $Input`, and $Input resolves to file 1, then the final command line, before execution, will be `test.sh "file 1"`. Therefore, do not put quotes around $Input or $Output.<br>**Note:** An example of a command line entry is *test.sh $Input $Output $0 $1 $2 $3 $4 $5 $6 $7 $8 $9.* This runs the shell script `test.sh` taking an input file, using ten parameters, and producing an output file. |
| Working Directory (workingDir) | Location of the directory to use for executing the command line process. Optional. Default is the current working directory of the JVM running CLA2Client.jar.<br><br>**CAUTION:**<br>**Using this adapter to call a Unix script modifies the directory path of the environment variable LD_LIBRARY_PATH. To keep your current path, your script should include either the LD_LIBRARY_PATH path or a reference to your .profile (which includes the LD_LIBRARY_PATH path).** |

| Field | Description |
|---|---|
| Turn on debugging messages? (cla2_debug) | Turn on debugging for this adapter instance? Valid values:<br><br>• Yes (true) – Logging is turned on and the messages are written to the system log.<br><br>• No (false) – Default.<br><br>**Note:** This turns on debugging for this specific adapter instance. These messages are logged in the system log in the *install_dir* logs directory. This parameter is read-only in the GPM.<br>**Note:** For backward compatibility, the CLA2 supports the Command Line adapter parameter cmdl_debug (at the business process level only). |
| Wait on the process to complete before continuing? (waitOnProcess) | Wait on the process to complete before continuing the business process. Valid values:<br><br>• Yes (true) – If the value is Yes, a status report is created if any stdout/stderr is generated by the process. If an error occurs while the service is processing output data, the advanced status contains the error message instead of the return code value.<br><br>• No (false)<br><br>**Note:** If **Use the output generated by the command line process** is set to Yes, the value of this parameter is assumed to be Yes because the service cannot use output if it does not wait for the process to complete. This parameter is read-only in the GPM. |
| Does this service start a business process? (bootstrap) | Whether the service starts a business process. Required. Valid values:<br><br>• Yes (true)<br><br>• No (false)<br><br>**Note:** This parameter is read-only in the GPM. |
| Business process (initialWorkFlowName) | Business process you want the Command Line Adapter 2 to start. This field is required only if you selected Yes in *Does this service start a business process?* . If you prefer to configure this parameter in the GPM, select Not Applicable.<br>**Note:** For backward compatibility, the CLA2 supports the Command Line adapter parameter initialWorkFlowId (at the business process level only). |
| Create Unique working directory | Command Line Adapter 2 creates a unique working directory for each invocation of a business process using the same Command Line Adapter 2 instance. Selecting this option ensures that the adapter instances do not overwrite each other when multiple files with the same name exist. |
| Document Storage Type (docStorageType) | Defines how the document is stored in the system. Required when the adapter starts a business process. Valid values:<br><br>• System Default – Default<br><br>• Database<br><br>• File System<br><br>**Note:** For more information on document storage types, see *Managing Services and Adapters*. |

| Field | Description |
|---|---|
| Run as User | 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<br><br>icon and select a user ID from the list. Valid value is any valid Sterling Integrator user ID.<br>**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. |
| 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 business process invoked by the Command Line Adapter 2. The Schedule field only displays as an option if *Does this service start a business process?* is set to Yes. Valid values:<br>• Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule.<br>• Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup.<br>• Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup.<br>• Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.<br>• Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |

| Field | Description |
|---|---|
| Does the command line process require an input file? (useInput) | Defines whether the command line process requires an input file? Required. Valid values:<br><br>• Yes (true) – The primary document of the current business process context is written out to the file system in the working directory and is used as input to the process. Default.<br><br>• No (false) – No file is written to disk even if a document exists in the business process context.<br><br>**Note:** This parameter is read-only in the GPM. |
| Input File Name (inputName) | Input file name, if the command line process requires an input file. Any occurrences of $Input in the command line are replaced with this name. Optional. If you leave this field blank, the default is the primary document name.<br>**Note:** It is important to have a unique input file name for all concurrently running instances of Command Line adapters. If more than one instance of the Command Line Adapter 2 can be executing at the same time, you must create a dynamic, unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine. |
| Delete input file after process completes? (inputDelete) | Defines whether the input file is deleted after the process completes? Valid values:<br><br>• Yes (true) – Default<br><br>• No (false)<br><br>**Note:** To delete the input file, **Wait on the process to complete before continuing?** must also be Yes. This parameter is read-only in the GPM. |
| Use the output generated by the command line process? (useOutput) | Use output generated by the command line process? Required. Valid values:<br><br>• Yes (true) – The adapter will attempt to read the output of the process. If bootstrapping a workflow, the file will become the primary document in the new workflow. If not bootstrapping, the file is collected and placed in ProcessData, not as the Primary Document. Default . For example,<br><br>`<assign name="Assign" to="PrimaryDocument"`<br>`    from="CLA2/document/@SCIObjectID"></assign>`<br><br>• No (false) – No file is read into the business process context even if one is generated by the command line process.<br><br>**Note:** This parameter is read-only in the GPM. |

| Field | Description |
|---|---|
| Output File Name (outputName) | Output file name, if you want to use the output generated by the command line process. Any occurrences of $Output in the command line are replaced with this name. Optional. If you leave this field blank, the default is the business process primary document name.<br>**Note:** It is important to have a unique output file name for all concurrently running instances of command line adapters. If more than one instance of the Command Line Adapter 2 can be executing at the same time, you must create a dynamic unique name to keep the instances from overwriting each other and causing the process to fail. This can be done by concatenating the current process ID on to a file's base name. This dynamic name may also need to be passed to the cmdLine. |
| Delete output file after process completes? (outputDelete) | Specifies whether the output file is deleted after it is collected?Valid values:<br>• Yes (true) – Default<br>• No (false)<br><br>**Note:** This parameter is read-only in the GPM. |

## GPM Configuration

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

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

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

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

| Field | Description |
|---|---|
| Config (participant name) | Name of the adapter configuration. Required. |
| env0 | An environment variable in the form name=value. Optional. Any value is valid. |
| env1 | An environment variable in the form name=value. Optional. Any value is valid. |
| env2 | An environment variable in the form name=value. Optional. Any value is valid. |
| env3 | An environment variable in the form name=value. Optional. Any value is valid. |
| env4 | An environment variable in the form name=value. Optional. Any value is valid. |
| env5 | An environment variable in the form name=value. Optional. Any value is valid. |
| env6 | An environment variable in the form name=value. Optional. Any value is valid. |
| env7 | An environment variable in the form name=value. Optional. Any value is valid. |
| env8 | An environment variable in the form name=value. Optional. Any value is valid. |
| env9 | An environment variable in the form name=value. Optional. Any value is valid. |
| keepPath | Normally, any path information is stripped off the filename to allow for platform independence. This parameter allows you to keep the entire path. Optional. Valid values:<br>• Yes – Path information is retained<br>• No – Path information is stripped off |
| parm0 | Resolves the $0 placeholder. Optional. Any value is valid. |
| parm1 | Resolves the $1 placeholder. Optional. Any value is valid. |
| parm2 | Resolves the $2 placeholder. Optional. Any value is valid. |
| parm3 | Resolves the $3 placeholder. Optional. Any value is valid. |
| parm4 | Resolves the $4 placeholder. Optional. Any value is valid. |
| parm5 | Resolves the $5 placeholder. Optional. Any value is valid. |
| parm6 | Resolves the $6 placeholder. Optional. Any value is valid. |
| parm7 | Resolves the $7 placeholder. Optional. Any value is valid. |
| parm8 | Resolves the $8 placeholder. Optional. Any value is valid. |
| parm9 | Resolves the $9 placeholder. Optional. Any value is valid. |

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

### Output from Adapter to Business Process

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

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

## Usage Examples

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

## Invoking the Command Line Adapter to Run a Shell Script

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

- When this example configuration is used, a shell script called "test.sh" (which resides in the /home directory) is run.
- The program requires the input filename as the first parameter, the output filename as the second parameter, and three other parameters.
- Because the useInput variable is set to true and the inputName variable is blank, the name of the primary document replaces the $Input placeholder.
- Because the useOutput variable is set to true and the outputName variable is blank, the $Output placeholder is replaced with the name of the primary document.
- If the document name in the workflow context is "data.txt" in this example, the command line becomes /home/test.sh data.txt data.txt VAR1 USER 10 at run-time.
- The name of the primary document is passed as the input file to the shell script program on the command line.
- The name of the primary document is passed as the output file to the shell script program on the command line.

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

## GPM Example

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



## Business Process Modeling Language (BPML) Example

The following example illustrates the same business process using BPML.

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

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

## Invoking the Command Line Adapter 2

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

To start the Command Line Adapter 2:

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

   **Note:** The CLA2Client.jar does not need to be moved if the remote version is on the same machine.
3. To start the remote adapter, enter:

   `java –jar CLA2Client.jar <port>` Example: `java –jar CLA2Client.jar 15699` The *port* can be any port that is not in use. This is the value to assign in the Remote Port field in the service configuration. See *Remote Port (remotePort)*. If you upgrade Sterling Integrator at any time, you will also need to use the corresponding new CLA2Client.jar file to avoid a "ClassConflict" error.

   **Note:** You may also run CLA2Client.jar with these commands:
   - (UNIX or iSeries)`./startCmdLine2.sh`
   - (Windows service) `startCLA2WindowsService.cmd`

## Stopping the Command Line Adapter 2

If Sterling Integrator is shut down with the (Windows) stopWindowsService.cmd or (UNIX and iSeries) hardstop.sh script, the Command Line Adapter 2 also shuts down.

You can also stop the Command Line Adapter 2 with these commands:
- (UNIX or iSeries) `./stopCmdLine2.sh`
- (Windows service) `stopCLA2WindowsService.cmd`

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

# Chapter 44. Compression Service

The following table provides an overview of the Compression service:

| System name | CompressionService |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | Provides the ability to compress (deflate) a document or group of documents in the business process context. It also provides the ability to decompress (inflate) a document. The following options are available with the Compression service:<br><br>• Compress all documents or just the primary document in a business process context.<br><br>• Decompress a .zip file and put a specified document into the primary document or decompress the file and start a business process for each document.<br><br>With the Compress option, parameters can be set to specify whether the files should be deflated or just stored. If the files are being compressed, parameters can be set to determine the level of compression. The decompression option can start a business process for each file, or it can put a specified document in the primary document. |
| Business usage | Used within a business process to perform compression or decompression of a document. |
| Usage example | If you have a zip file in the primary document, you can use this service to decompress the file and start a business process for each document found. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | No |
| Application requirements | No |
| Initiates business processes? | If decompressing files, and Decompress_result is set to Start_bpml, child business processes are started for each document in compressed file. |
| Invocation | Runs as part of a business process. |
| Business process context considerations | This service uses the primary document and can also compress other documents in the business process context. |
| Returned status values | Returned status values:<br><br>• Success – Compression or decompression was successful.<br><br>• Error – Errors were encountered during compression or decompression. The report contained in the Workflow Context Status report should be consulted for further detail. |

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

## How the Compression Service Works

The Compression service provides two options:
- Compression
- Decompression

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

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

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

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

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

Compression

One or more documents in business process context → Files are compressed → Compressed file is put into business process context (primary document)



Decompression

Zip file (contains one or more docs) → Zip file is decompressed by Compression service →
Option 1: Starts business process
Option 2: Puts a specific file in the business process context

## Business Process Example

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

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

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

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

## Implementing the Compression Service

To implement the Compression service, complete the following tasks:

1. Configure the pre-installed copy of the Compression service, or create a new configuration. For information, see *Configuring the Compression Service*.

2. Use the Compression service in a business process.

## Configuring the Compression Service

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

| Field | Description |
|-------|-------------|
| Config | Name of the service configuration. Required. |
| bpml_name | Name of the business process to start. Required if you selected Start_bpml for Decompress_result field. Valid value is a string. Decompression parameter only. |
| compressed_filename | What to name the compressed file. Valid value is a string: file name with extension. Default is CompressedFile<date/timestamp>.zip. Date/Timestamp is in the format CCYYMMDDHHMMSSMMS. Compression parameter only. |

| Field | Description |
|---|---|
| compression_action | Whether to compress or decompress documents. Valid values are Compress and Decompress. |
| compression_level | Level of compression. Valid values are 0 - 9 (default is 0). Compression parameter only. |
| compression_type | Compression method. Valid values are Deflate and Store (default is Deflate). Compression parameter only. |
| decompress_result | What to do with decompressed files. Valid values are Start a business process and Put a specific document in the primary document. Decompression parameter only. Required. |
| doc_to_compress | Whether to compress all docs or just the primary doc. Valid values are All and Primary Doc (default is All). Compression parameter only. |
| file_name | Name of file to put in primary document. Valid value is a string: name of file with extension. Defaults to first document in the list. Decompression, primary_doc parameter only. **Note:** When decompressing: if you choose to put a specific document in the primary document area, but fail to specify a filename when multiple files exist in the compressed input file, Sterling Integrator will select the first of the files present, as determined by the compression utility that produced the compressed file. |

# Chapter 45. Connect:Direct Server Adapter

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

| System Name | Connect:Direct Server Adapter (CDSA) |
|---|---|
| Graphical Process Modeler (GPM) category | Not configurable in the GPM |
| Description | This adapter receives and processes requests from remote Connect:Direct nodes. |
| Business usage | Use this adapter to:<br><br>• Mediate requests originating from a business process and going to an external trading partner using a (non-application) Connect:Direct node or another Connect:Direct Server adapter<br><br>• Accept requests from a Connect:Direct node (or another Connect:Direct Server adapter) to copy to or from a mailbox or business process<br><br>• Forward requests from one remote Connect:Direct node to another |
| Usage example | A trading partner submits a Connect:Direct Process on their local system. The Process initiates a session with the application Connect:Direct node and requests to copy a file. The Connect:Direct Server adapter allows the session and responds to the copy request. |
| Preconfigured? | No preconfigured copy of the adapter is provided. |
| Requires third party files? | The following files external to your application are required:<br><br>• Connect:Direct Server<br><br>• Certicom SSL Library (included with your application)<br><br>• JZLIB (included with your application)<br><br>• CDJava.jar (build 81) (included with your application) |
| Platform availability | All supported application platforms |

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

| | |
|---|---|
| Restrictions | The Connect:Direct Server adapter includes parameters to govern checkpoint/restart behavior if either the application or the remote node fails. If the remote Connect:Direct Server or the network fails during a copy operation, the adapter goes into a retry mode using the values specified at configuration. The adapter waits the configured amount of time and then resumes the copy. The length of time the adapter waits to resume the copy is impacted by three values:<br><br>• Number of Retry attempts, as specified in the adapter configuration.<br><br>• Interval between Retry attempts, as specified in the adapter configuration.<br><br>• Frequency defined by the scheduled system business process BPExpirator. BPExpirator is used in the Wait service. The Wait service is used when a system failure interrupts a Connect:Direct copy operation.<br><br>Whichever value is longer governs behavior. By coordinating the adapter retry interval and the BPExpirator schedule, you can configure a window of time appropriate for your needs. Your application enables automatic restart when the recovery level is set to Auto-Resume. Checkpoint restart is only enabled for file system and mailbox documents. All other restarted copy operations are started from the beginning of the document.<br><br>**CAUTION:**<br>**If you add or delete a node or modify node parameters for any of the adapter's netmap members, disable and reenable the Connect:Direct Server adapter. Omitting this step can cause a significant delay (up until the next day) before configuration changes to netmap members are recognized.** |
| Persistence level | None |
| Testing considerations | Debug information for this adapter can be found in the Connect:Direct Server adapter log files. |

### Implementing the Connect:Direct Server Adapter

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

1. Collect information about any remote Connect:Direct servers that will communicate with this adapter. You will use this information during configuration of the adapter. For information, see *Configuring the Connect:Direct Server Adapter*.

   If planning to use a Perimeter server with this adapter, configure the Perimeter server. For information, see *Working with Perimeter Servers*.

2. Add the application as a node to the Connect:Direct network map.

3. Create a Connect:Direct Server adapter configuration. For information, see *Managing Services and Adapters*.

4. Configure the Connect:Direct Server adapter. For information, see *Configuring the Connect:Direct Server Adapter*.

5. If you add or delete a node or modify node parameters for any of the adapter's netmap members, disable and reenable the Connect:Direct server adapter.

**Note:** If this step is omitted, there may be a significant delay (up until the next day) before configuration changes to netmap members are recognized.

6. If using the upload and download parameters (proxy or user repositories), configure these property file. See *Configuring CDSA Property Files*.
   - cdinterop-user-records.properties
   - cdinterop-proxy-records.properties

   **Note:** The use of cdinterop-proxy-records.properties will also require entries in cdinterop-spoe-policy.properties and cdinterop-spoe-auth.properties.

7. Create configurations of any Connect:Direct server services needed to use this adapter as a PNODE. See *Related services* in the Overview table for a list of these services.

8. Use the Connect:Direct Server services in a business process to use the adapter as a PNODE.

## Configuring the Connect:Direct Server Adapter

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

| Field | Description |
|---|---|
| Name | Unique and meaningful name for the adapter configuration. Required. |
| Description | Meaningful description for the adapter configuration, for reference purposes. Required. |
| Environment | (Clustered environments only.) Nodes available for this adapter to run on. **Note:** "All" is not a valid value. The netmap expects a specific node at a specific host. |
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br>• None – You do not want to include this configuration in a group at this time (default)<br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br>• 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.<br>For more information about service groups, see *Managing Services and Adapters*. |
| Connect:Direct Server Node Name | Name that will be used to identify this Connect:Direct Server adapter. Maximum 16 characters. Must match the Netmap entry for server communicating with. Required. |
| Connect:Direct Perimeter Services Option | Perimeter servicer to be used. Valid values:<br>• node1 & local (Default) – Local install<br>• Any perimeter server(s) configured in the application for the node selected for Environment. |

| Field | Description |
|---|---|
| Connect Direct Server Port | TCP/IP port number that Connect:Direct monitors for requests from remote nodes. Required.<br>**Note:** If your application is installed on a UNIX operating system, do not select ports in the range 1-1024. These are normally reserved by the operating system. |
| Firewall Ports | Ports used by the adapter for making outbound connections through the firewall. Optional. Valid values are hyphenated port range, a comma-delimited list, or a combination of both. |
| Max locally initiated (pnode) sessions allowed | Maximum number of sessions the Connect:Direct Server adapter may have active at any point of time. Required. May be overridden at the node level. If a node-level value is configured, the effective session limit is the smaller of two values: the limit for the adapter and the session limit for the remote node. Valid value: Any number through 9999. Minimum value is 1. Default is 5. |
| Max remotely initiated (snode) sessions allowed | Maximum number of sessions from remote nodes that may be active at any point of time. Required. May be overridden at the node level. If a node-level value is configured, the effective session limit is the smaller of two values: the limit for the adapter and the session limit for the remote node. Valid value: Any number through 9999. Minimum value is 1. Default is 5. |
| Document Storage Type | Defines how the document will be stored in the system. Required. Valid values are:<br>• System Default<br>• Database<br>• File System (default) |
| Netmap Check | Whether the node netmap should be checked to authorize sessions. Required. Valid values are:<br>• Check both node name and IP address – Searches the CDSA network map for an entry corresponding to the node name of the remote client. If one exists, the IP address is obtained and compared to the IP address of the remote node making the request.<br>• Check node name only – Searches the CDSA network map for an entry corresponding to the node name of the remote client.<br>• No (default) – Disables netmap validation.<br>**Note:** Connect:Direct Server adapters that were configured before availability of the "Check node name only" option are not affected. The previous "Yes" value corresponds to "Check both node name and IP address." |
| Buffer Size for Copy | Size (in bytes) of the internal buffer used when accessing a file. Optional. Valid values are:<br>• 4096<br>• 8192<br>• 16384<br>• 32768 (Default) |

| Field | Description |
|---|---|
| Number of Retry Attempts for Checkpoint-based copy restart | Maximum number of attempts for checkpoint restart retry. Required. Valid value is any number through 9999. Default is 3.<br>**Note:** Checkpoint restart is only enabled for file system-based document, outbound (from your application) data transfers. (If your application is the Pnode, this means business processes using the CopyTo service. If your application is the Snode, this means the remote Connect:Direct Server copies documents from your application to the remote node.) |
| Interval between Retry attempts (minutes) | Amount of time to wait between checkpoint restart retry attempts. Required. Valid value: Any number through 99999999. Default is one minute.<br>**Note:** If the Schedule_BPExpirator business process is configured to wait longer than the adapter retry interval indicates, the BPExpirator value will govern behavior. The default value for BPExpirator is 15 minutes. If the adapter retry interval is longer, the adapter value will govern behavior.<br>**Note:** Checkpoint restart is only enabled for file system-based and mailbox-based documents, outbound (from your application) data transfers. |
| Number of short-term session retry attempts | Specifies the maximum number of short-term session retry attempts that should be made when failures to connect occur. This setting can be overridden by the value specified for ShortTermMaxRetries in the Connect:Direct Server Begin Session service. Valid value: Any number up through 9999. Default is 0. Required. Short-term retry activity is logged (if logging enabled). |
| Interval between short-term session retry attempts (seconds) | Specifies the amount of time to wait between short-term session establishment attempts. This performs successive retires within a shorter period of time than is available with the Interval between Retry attempts (minutes) settings. This setting can be overridden by the value specified for ShortTermRetryInterval in the Connect Direct Server Begin Session service. Valid value: Any number through 99999999. Default is 5. Required. Short-term retry activity is logged (if logging enabled). |
| Number of retry attempts for establishing a session | Specifies the maximum number of attempts for creating a session with a Connect:Direct node. This setting can be overridden by the value specified for Begin Session Max Retries in the Connect:Direct Server Begin Session service. Valid value: Any number up to 9999. Default is 0. Required. Short-term retry activity is logged (if logging enabled). The following errors fail without any retries:<br>• Invalid required parameter<br>• Missing required parameter<br>• Adapter not found<br>• Remote node not in the netmap<br>Connect:Direct unable to locate local server |
| Interval between session establishment attempts (minutes) | Time, in minutes, to wait between session establishment attempts. This setting can be overridden by the value specified for Begin Session Retry Interval in the Connect Direct Server Begin Session service. Any number through 99999999. Required. |

| Field | Description |
|---|---|
| Max Session Establishment Timeout value in Seconds | Amount of time to wait before timing out an attempted connection with the Perimeter Server. Valid value is any number through 99999999. Default is 600. Required. |
| Max Socket Read Timeout value in Seconds: | Specifies the amount of time to wait for the remote Connect:Direct Server to respond to any message sent by the Connect:Direct Server adapter. Required. Valid value is any number through 99999999. Default is 90. When using a Perimeter server, the minimum is 90. If a value less than 90 is entered, 90 is applied. |
| Server Start Option | Whether the server will be Cold or Warm started. A Warm start maintains existing restart information. A Cold start resets the restart information. Default is Warm. Required. |
| RunTask Business Process Name | Select a business process to be run when the Connect:Direct Server adapter receives a RunJob request. Optional. This option is primarily for use with business processes that forward requests from one Connect:Direct node to another. You must write the business process to do the forwarding. **Note:** For information about the forwarding option, see *Business Process Examples*. |
| RunJob Business Process Name | Select a business process to be run when the Connect:Direct Server adapter receives a RunJob request. Optional. This option is primarily for use with business processes that forward requests from one Connect:Direct node to another. You must write the business process to do the forwarding. **Note:** For information about the forwarding option, see *Business Process Examples*. |
| Max Run Task Forwarding Timeout value in Seconds: | The amount of time (in seconds) to wait before timing out a session that forwards a RunTask request. Required. Default is 300. Valid value is any natural number. |
| Max Run Job Forwarding Timeout value in Seconds: | The amount of time (in seconds) to wait before timing out a session that forwards a RunJob request. Required. Default is 30. Valid value is any natural number. |
| Extractable Count | The number of times the message can be extracted. Optional. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer. |
| Extractable For | The length of time (in days, hours and minutes) the message can be extracted. Optional. Cannot be specified in conjunction with Extractable or Extractable Count. Format is *ddhhmm*. |
| Extractable | A yes or no value indicating if this message can be extracted. Optional. Cannot be specified in conjunction with Extractable Count or Extractable For. Valid values are Yes and No. |
| Encryption using Secure+ | Whether Secure+ is enabled. Required. Valid values are Enabled and Disabled. Default is Disabled. **Note:** The following four fields are displayed only if Enabled is selected. |
| Enable Netmap Node Override | If Yes, specifies that remote node netmap security settings override the adapter security settings. Required. Valid values are Yes and No. Default is No. |

| Field | Description |
|---|---|
| CA Certificates | Drop-down menu that contains a list of trusted Certificate Authority public certificates. Only displayed if "Encryption using Secure+" is enabled. Required if Encryption using Secure+ is enabled. Valid values are all CA certificates in this installation of your application. |
| System Certificate | List of private keys and public certificates signed by the trading partner trusted certificate authority. This represents the certificate for the local node (this adapter). Only displayed if Encryption using Secure+ is enabled. Required if Encryption using Secure+ is enabled. Valid values are all system certificates in this installation of your application. |
| Certificate Common Name | Value to be compared with the Common Name (CN) field from the remote node's certificate during TLS/SSL handshake in order to enhance certificate authentication. Only displayed if Encryption using Secure+ is enabled. May be overridden at the node level. **Note:** To see the actual CNs being compared during the handshake, set the Perimeter Services logging level to ALL. |
| Cipher Suites | Indicates cipher suites available to support varying encryption levels. Multiple selections allowed. Selections can be rank ordered. Only displayed if "Encryption using Secure+" is enabled. Required if "Encryption using Secure+" is enabled. Valid values are all available cipher suites in this installation of your application. |
| SSL or TLS | Indicates if Secure+ is enabled using SSL or TLS. Only displayed if "Encryption using Secure+" is enabled. Required if "Encryption using Secure+" is enabled. Valid values are SSL and TLS. Default is SSL. |
| Require Client Authentication: | Indicates whether authentication occurs at the client level. Valid values are:YesNo |

## CDSA-related Property Files

These property files might be used in conjunction with the Connect:Direct Server adapter.

In clustered environments, the properties files on every node should be modified. Secondary nodes will not be aware of properties files changes on the primary node.

| Property File | Function |
|---|---|
| cdinterop-spoe-auth. properties | Used to determine the specific proxy user(s) on Sterling Integrator. That is, it maps the Connect:Direct ID to the account ID on Sterling Integrator |
| cdinterop-spoe-policy. properties | Enables a proxy user relationship to exist between Connect:Direct and Sterling Integrator and sets a precedence for it (default: address, domain, account). It also determines if an snodeid may be present. |

| Property File | Function |
|---|---|
| cdinterop-user-records. properties | Maps external users/nodes to upload and download paths for storing incoming files or retrieving an outgoing file. This allows the CDSA to process files despite restrictions such as limited file name length.<br>**Note:** These upload/download paths were formerly maintained in the business process as part of the document name. Now, only the file name is included in the business process or the CD process, and the key word mailbox or business process (as well as any subpath for the mailbox) is in the properties file.The cdinterop-user-records.properties file can be setup to map with ID only, with ID and node name, or node name only. If used in conjunction with the SPOE settings, the mapping is to the proxied user from SPOE. Examples:<br><br>cdproxy1[CDSA1]download=/mailbox/sub1<br><br>cdproxy1[CDSA1]upload=/mailbox/sub1<br><br>cdproxy2/upload=/mailbox/sub2<br><br>cdproxy2/download=/mailbox/sub2<br><br>cdproxy3/download=/businessprocess/cdfsa |
| cdinterop.proxy.records.properties | Assigns an upload or download document path to a proxied user ID |

## Configuring CDSA Property Files

Configuration of a Connect:Direct Server adapter may include configuring the cdinterop.user.records.properties file, the cdinterop.proxy.records.properties, or both. Entries may be added, modified, or deleted in the properties files at any time, without starting or stopping either the application or the adapter. Changes take effect within a few seconds.

Editing the cdinterop.proxy.records.properties file requires entries in cdinterop-spoe-auth.properties and cdinterop-spoe-policy.properties.

### Add or Modify User or Proxy Entries Before Startup

1. Add or modify entries in the user or proxy repository file, using the appropriate syntax:
   - To edit cdinterop-user-records.properties: *userid* ([*cdsa_name*] | / ) upload | download = *gis-path-prefix* See the cdinterop-user-records.properties file documentation for parameter definitions.
   - To edit cdinterop-proxy-records.properties: *external-id* / upload | download = *gis-path-prefix* See the cdinterop-user-records.properties file documentation for parameter definitions.
2. Start the application.

## Add or Modify User or Proxy Entries While the Application is Running

Entries may be added, modified, or deleted in cdinterop-user-records.properties and cdinterop-proxy-records.properties while the application is running. The Connect:Direct Server adapter does not need to be stopped and restarted.

This procedure assumes that both properties files are being modified. Edit and commit one or both, as needed.

1. Make a copy of the cdinterop-user-records.properties file:

```
$ cd <install_dir>/properties
$ cp cdinterop-user-records.properties
 cdinterop-user-records.properties.copy
```

1. Use your preferred text editor to edit the file copy, and save. Example: `$ vi cdinterop-user-records.properties.copy`

2. Commit the changes by copying the modified file back to the original filename: `$ cp cdinterop-user-records.properties.copy cdinterop-user-records.properties`

3. Make a copy of the cdinterop-proxy-records.properties file: `$ cp cdinterop-proxy-records.properties cdinterop-proxy-records.properties.copy`

4. Use your preferred text editor to edit the file copy, and save. Example: `$ vi cdinterop-proxy-records.properties.copy`

5. Commit the changes by copying the modified file back to the original filename: `$ cp cdinterop-user-records.properties.copy cdinterop-user-records.properties` Your changes will take effect within a few seconds.

## Search Order for Download/Upload Parameters

When a Connect:Direct Server adapter requests to connect to a remote Connect:Direct server node, the value of the upload or download parameter is determined as follows:

| When a CDSA requests a connection in order to: | The upload/download parameter value is determined in this way: |
|---|---|
| Send a file to be stored on the Connect:Direct server node | Upload/download parameter values do not apply. |
| Request a file to be retrieved from the C:D server node. | The download parameter definition will be used if one is defined. The CDSA determines the value of the parameter in the following manner: <br> 1. The User Record Repository is searched for a matching definition. <br> 2. If a match is found, the remote file specification is appended to the matched parameter's value to provide a fully qualified local filename. |

When a remote Connect:Direct server node requests a CDSA connection, the value of the upload or download parameter is determined as follows:

| When a remote note requests a connection in order to: | The upload/download parameter value is determined in this way: |
|---|---|
| Request a file to be retrieved from the CDSA platform<br><br>OR<br><br>Send a file to be stored on the CDSA platform | The upload or download parameter definition is used, if defined. The CDSA determines the value of the parameter in the following manner:<br><br>1. If the remote user ID is a proxy ID for a local Sterling Integrator user account (SPOE policy is enabled) then the Proxy Record Repository is searched for a matching definition.<br><br>2. If no match is found, or if the remote ID is not proxied (SPOE policy is not enabled), then a search is performed on the User Record Repository using the local GIS user account ID. If SPOE policy is enabled, the local Sterling Integrator user account ID associated with the proxied remote user is used for this search.<br><br>3. If a match is found in either case above, the remote file specification is appended to the matched parameter's value to provide a fully qualified filename. |

## Reserved Words

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

- `businessprocess`
- `mailbox`

These must be used at the beginning of the path for the target (or source) of the copy/submit in your application. For example, the following could be used in the TO or FROM clause of a COPY statement in a Connect:Direct process script:

`/businessprocess/myspecialBPname/targetdocName`

or

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

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

## Business Process Examples

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

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

```
<process name=" CopyToExample">
  <sequence name="CDServerCopyToGIS">
<sequence name="CDServerCopyToGIS">
```

```
        <!-- Begin Server Session -->
        <operation name="Connect:Direct Server Begin Session Service">
          <participant name="CDServerBeginSession"/>
          <output message="BeginSession">
            <assign to="." from="*"></assign>
            <assign to="LocalCDNodeName">TEST1_CDSERVER</assign>
            <assign to="RemoteCDNodeName">REMOTECDNODE1</assign>
            <assign to="RemoteUserId">user1</assign>
            <assign to="RemotePasswd">password1</assign>
          </output>
          <input message="inmsg">
            <assign to="BeginSessionResults" from="*"></assign>
          </input>
        </operation>
          <operation name="Connect:Direct Server Copy To Service">
<operation name="Connect:Direct Server Copy To Service">
          <participant name="CDServerCopyTo"/>
          <output message="CopyTo">
            <assign to="." from="*"></assign>
            <assign to="SessionToken"
                    from="//BeginSessionResults/SessionToken/node()">
</assign>
            <assign to="RemoteFileName">sample.txt</assign>
            <assign to="BinaryMode">no</assign>
            <assign to="Sysopts">datatype(text) xlate(no)</assign>
             </output></output>
          <input message="inmsg">
            <assign to="CopyToResults" from="*"></assign>
          </input>
        </operation>
<!-- End Server Session -->
        <operation name="Connect:Direct Server End Session Service">
          <participant name="CDServerEndSession"/>
          <output message="EndSession">
            <assign to="." from="*"></assign>
            <assign to="SessionToken"
                    from="//BeginSessionResults/SessionToken/node()">
</assign>
          </output>
          <input message="inmsg">
            <assign to="EndSessionResults" from="*"></assign>
          </input>
        </operation>
  </sequence></sequence>
</process>
```

## Forwarding a Request from One Connect:Direct Node to Another

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

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

```
<operation name="Connect:Direct Server Run Job Service">
     <participant name="CDServerRunJob"/>
     <output message="RunJob">
       <assign to="." from="*"></assign>
       <assign to="SessionToken"
               from="//BeginSessionResults/SessionToken/node()"></assign>
       <assign to="Sysopts">AAAA1.jcl</assign>
     </output>
     <input message="inmsg">
               <assign to="Run_Job_Results" from="*"></assign>
     </input>
   </operation>
<operation name="CONNECT:Direct Server BP Response Service">
        <participant name="CDServerTestBPResponse1"/>
        <output message="BPResponseParams">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="BPResponseResults" from="*"</assign>>
<assign to="BPResponseResults" from="*"></assign>
        </input>
   </operation>
   <operation name="Connect:Direct Server End Session Service">
<operation name="Connect:Direct Server End Session Service">
     <participant name="CDServerEndSession"/>
     <output message="EndSession">
       <assign to="." from="*"></assign>
       <assign to="SessionToken"
               from="//BeginSessionResults/SessionToken/node()"></assign>
     </output>
     <input message="inmsg">
       <assign to="OnFaultHandler" from="*"></assign>
     </input>
   </operation>
<!-- Cleanup when bad things happen (exceptions) -->
   <onFault>
   <sequence name="OnFault">
<operation name="CONNECT:Direct Server BP Response Service">
        <participant name="CDServerTestBPResponse1"/>
        <output message="BPResponseParams">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="BPResponseResults" from="*"></assign>
        </input>
   </operation>
   <!-- End the Session which was begun before the fault code -->
<!-- End the Session which was begun before the fault code -->
   <operation name="Connect:Direct Server End Session Service">
     <participant name="CDServerEndSession"/>
       <output message="EndSession">
   <assign to="." from="*"></assign>
     <assign to="SessionToken"
               from="//BeginSessionResults/SessionToken/node()"></assign>
     </output>
     <input message="inmsg">
       <assign to="EndSessionResults" from="*"></assign>
     </input>
   </operation>
   </sequence></sequence>
   </onFault>
   </sequence></sequence>
</process>
```

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

```
<process name = "CDInterop_testRunTaskForward">
  <sequence name = "CDBeginSession">
    <operation name="Connect:Direct Server Begin Session Service">
      <participant name="CDServerBeginSession"/>
      <output message="BeginSession">
        <assign to="." from="*"></assign>
        <assign to="LocalCDNodeName">aaaa.bbbb</assign>
        <assign to="RemoteCDNodeName">ZZ.AAA.ZZZZZ</assign>
        <assign to="RemoteUserId">username</assign>
        <assign to="RemotePasswd">password</assign>
      </output>
      <input message="inmsg">
        <assign to="BeginSessionResults" from="*"></assign>
      </input>
    </operation>
    <operation name="Connect:Direct Server Run Task Service">
<operation name="Connect:Direct Server Run Task Service">
      <participant name="CDServerRunTask"/>
      <output message="RunTask">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
                from="//BeginSessionResults/SessionToken/node()"></assign>
        <assign to="Sysopts">AAAAA</assign>
      </output>
      <input message="inmsg">
        <assign to="Run_TASK_Results" from="*"></assign>
      </input>
    </operation>
    <operation name="CONNECT:Direct Server BP Response Service">
<operation name="CONNECT:Direct Server BP Response Service">
      <participant name="CDServerTestBPResponse1"/>
      <output message="BPResponseParams">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="BPResponseResults" from="*"></assign>
      </input>
    </operation>
    <operation name="Connect:Direct Server End Session Service">
<operation name="Connect:Direct Server End Session Service">
      <participant name="CDServerEndSession"/>
      <output message="EndSession">
        <assign to="." from="*"></assign>
        <assign to="SessionToken"
                from="//BeginSessionResults/SessionToken/node()"></assign>
      </output>
      <input message="inmsg">
        <assign to="OnFaultHandler" from="*"></assign>
      </input>
    </operation>
<!-- Cleanup when bad things happen (exceptions) -->
    <onFault>
    <sequence name="OnFault">
<operation name="CONNECT:Direct Server BP Response Service">
      <participant name="CDServerTestBPResponse1"/>
      <output message="BPResponseParams">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="BPResponseResults" from="*"></assign>
      </input>
    </operation>
  <!-- End the Session which was begun before the fault code -->
    <operation name="Connect:Direct Server End Session Service">
      <participant name="CDServerEndSession"/>
      <output message="EndSession">
        <assign to="." from="*"></assign>
```

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

## Activity Types for This Service

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

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

# Chapter 46. CPA Lookup Service

The following table provides an overview of the CPA Lookup service:

| System name | CPALookup |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | Retrieves the relevant Collaborative Partner Agreement (CPA) information from an XML file based on an input CPA ID. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Returned status values | Returned status values:<br>• Basic status Success – Transformation was successful.<br>• Basic status Error – Errors were encountered during transformation or transformation could not be performed. The report contained in the Workflow Context Status report should be consulted for further detail. |
| Restrictions | The CPA Lookup service can only read information stored in cpa-store.xml. |
| Testing considerations | To test the CPALookup service, make sure you have set the location of CPA Store correctly. Create a sample business process and execute it. The most frequent problems encountered are:<br>• Invalid CPA Store location<br>• Non existent cpa-id, service name, service type and action name in CPA Store<br>• Adapter is not active. |

## Implementing the CPA Lookup Service

To implement the CPA Lookup service, complete the following tasks:

1. Use the pre-installed copy of the CPA Lookup service, or create a new configuration. See *Managing Services and Adapters*.
2. Use the CPA Lookup service in a business process.

## Configuring the CPA Lookup Service

There is no configuration necessary for the CPA Lookup service.

# Chapter 47. Certificate Revocation List (CRL) Processing Service

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

| System name | CRLProcessingService |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | This service processes a version 2 digital Certificate Revocation List (CRL). Specifically, this service:<br>• Marks those certificates in the Sterling Integrator database which appear on the list as revoked or held<br>• Checks the cache and notifies you that a specific certificate is revoked when an attempt is made to check in that system or trusted certificate.<br>**Note:** The notification only occurs if you enable CRL caching in Sterling Integrator. |
| Business usage | Use this service to confirm that your business processes correctly authenticate your trading partners. This service confirms that other Sterling Integrator services that request digital certificates are only given valid certificates. Valid certificates are those that have not been revoked or held. Also use this service if you want to be notified when an attempt is made to check in a revoked certificate. |
| Usage example | The CRL Processing service should be placed in a scheduled business process that downloads CRLs at the required interval. (The required interval varies by authority and trading community.)<br>**Note:** The CRL does not contain certificates; it lists those certificates that should be revoked. For more information, see *How the CRL Processing Service Works.* |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The HTTP Client adapter and the LDAP adapter are often used in conjunction with this service because these adapters can be used to download the CRL. This service confirms that other services which request the use of digital certificates are given them only if the certificate has not been revoked or is being held. |
| Application requirements | The list passed to this service must be formatted according to the CRL version 2 specification published by the Internet Engineering Task Force (IETF) in document http://www.ietf.org/rfc/rfc3280.txt. |
| Initiates business processes? | No |
| Invocation | This service must be preceded by a service or adapter that can collect the CRL and hand it off to this one. In addition, the entire chain of authorization for the CRL must be in the Sterling Integrator CA certificate store before this service can run. |

| | |
|---|---|
| Business process context considerations | This service processes the CRL; it does not download the CRL. Before processing the CRL, this service attempts to verify the signature on the CRL. This service can verify the CRL only if the following certificates are in the Sterling Integrator certificate store:<br>• The certificate used to sign the CRL<br>• Any intermediate certificates between that certificate and the CA root certificate<br>• The CA root issuing certificate |
| Returned status values | Returned status values:<br>• Success – The CRL processed correctly.<br>• Error – The CRL did not process correctly.<br>For a complete description of the errors, see the log file. |
| Restrictions | This service only processes the following CRL critical extensions:<br>• crlNumber<br>• baseCRLNumber<br>• IssuingDistributionPoints. |
| Persistence level | None |
| Testing considerations | Enable the ShowTranscripts parameter in the CRL Processing service configuration. When enabled, operations on each certificate in the list are recorded. |

## How the CRL Processing Service Works

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

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

1. It verifies the CRL using a certificate in the CA certificate store in the Sterling Integrator database.

2. Once verified, it checks the CRL expiration date. If the CRL is expired, the default behavior is for the service to fail. You can allow the service to continue by changing the FailOnExpiredCRL parameter to false.

3. The service then begins to process the CRL. If the CRL has a critical extension that the application does not understand, the default behavior for this service is to fail. You can allow the service to continue by changing the FailOnUnsupportedCRLCritExtns parameter to false.

4. For each entry in the CRL, the service looks in the trusted certificate and system certificate tables for a corresponding certificate entry.

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

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

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

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

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

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

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

- If there is no corresponding CRL of that type and issuer, the service simply adds the CRL to the database.
- If there is an existing CRL, the service uses the CRL number to add the most recent CRL to the database.

  - If the CRL numbers are identical, the service uses the thisUpdate date in the CRLs to determine which is the most recent.

  - If there are no CRL numbers, the thisUpdate date is used.

6. The service updates the CRL cache.

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

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

## Implementing the CRL Processing Service

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

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

## Configuring the CRL Processing Service

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

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

| Field | Description |
| --- | --- |
| Config | Name of the service configuration. Required. |

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

## Output from Business Process to Service

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

## Business Process Example

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

```
<process name="1CRL">
  <sequence name="optional">
    <operation name="One">
      <participant name="1CRL"/>
        <output message="noopout">
         <assign to="." from="*"></assign>
         <assign to="ShowTranscripts">true</assign>
         <assign to="FailOnExpiredCRL">false</assign>
         <assign to="FailOnUnsupportedCRLCritExtns">false</assign>
        </output>
        <input message="noopin">
         <assign to="." from="*"></assign>
        </input>
    </operation>
  </sequence>
</process>
```

# Chapter 48. Cryptographic Message Service

The following table provides an overview of the Cryptographic Message service:

| | |
|---|---|
| Service name | Cryptographic Message Service |
| System name | CryptoMsgService |
| Graphical Process Modeler (GPM) category | All Services |
| Description | Builds and parses cryptographic messages in SMIME, PEM, or DER format. |
| Business usage | The Cryptographic Message Service allows users to build and parse cryptographic messages in SMIME, PEM, or DER format. |
| Usage example | A business process that needs to create or parse the content in a cryptographic message in SMIME, PEM, or DER format can invoke this service by passing the required parameters. Cryptographic messages must follow either Cryptographic Message Syntax or PKCS#7 specification. |
| Preconfigured? | The Cryptographic Message service should be installed and deployed before it is invoked. However, configuration parameters are not required. |
| Requires third party files? | Yes. Requires Certicom sbgsepkij3.3 jars. This is preloaded in Sterling Integrator. |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | No |
| Application requirements | No |
| Initiates business processes? | No. This service does not initiate business process. |
| Invocation | Yes. Runs as a service within a business process. |
| Returned status values | Values:<br>• buildResponse - If an exception is thrown during build process, the "exception-message" node is returned to ProcessData with the exception message.<br>• parseResponse - If an exception is thrown during parse process, the "exception-message" node is returned to ProcessData with the exception message. |
| Restrictions | None |
| Testing considerations | Considerations<br>• You should use the right certificates for signing or encryption/decryption.<br>• If you receive an error with the condition that certificates used for signing or decrypting are not created with a storepass value of integrator and are created with a keypass value of integrator, see your system administrator. |

## How the Cryptographic Message Service Works

Cryptographic Message Service (CMS) builds and parses secure messages in Secure MIME (SMIME), Distinguished Encoding Rules (DER), or Privacy Enhanced Email (PEM) format.

The security features of CMS are digital signature and encryption. The Digital signature feature provides authentication, message integrity, and non-denial with proof or origin whereas encryption provides data privacy.

The CMS supports two cryptographic message syntaxes. They are CMS and PKCS#7. If you are building outbound message syntax, you have to indicate the cryptographic message syntax as either one of them. The PKCS#7 uses non-streaming API to handle message building and has limitations to process large files whereas the CMS uses streaming API and has the capability to process large files. If you are parsing an inbound cryptographic message, there is no need to indicate your choice as CMS uses streaming API to parse either PKCS#7 or CMS messages.

## Implementing the Cryptographic Message Service

To implement the Cryptographic Message service for use in a business process, complete the following tasks:

1. Create a configuration of the Cryptographic Message service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Cryptographic Message Service*.
2. Specify field settings for the service configuration in the Sterling Integrator Admin Console and in the GPM as necessary. For information, see *Configuring the Cryptographic Message Service*.
3. Use the Cryptographic Message service in a business process.

## System Administrator Tasks

The following procedures describe the system administrator tasks for the cryptographic message service.

## Importing a keyCert

1. Login to Sterling Integrator.
2. Select **Trading Partner** -> **Digital Certificates** -> **System**.
3. Select **Key Certificate** under Check in.
4. Enter the Certificate Name and Private Key Password.
5. Select the certificate and assign an alias to it.
6. Review and click **Finish**. You can use this certificate in your BPML associated with the appropriate field (signingCert or decryptCert).

## Importing a Public Certificate

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 **Next**.

6. Review and click **Finish**. You can use this certificate in your BPML associated with the appropriate field (encryptCert or sigVerifyCert).

## Configuring the Cryptographic Message Service

You can create one service instance for building and parsing cryptographic messages. You can configure the service in Sterling Integrator and also in the GPM.

To configure the Cryptographic Message service, you must specify settings for the following fields:

**Note:** Any field values passed from a prior service can override any of configured fields for this service.

| Field | Description |
|-------|-------------|
| Name | Unique and meaningful name for the adapter configuration. Required. |
| Description | Meaningful description for the adapter configuration, for reference purposes. Required. |
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br>• None - You do not want to include this configuration in a group at this time (default)<br>• Create New Group - You can enter a name for a new group in this field, which is then created along with this configuration.<br>• 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.<br><br>For more information about service groups see *Managing Services and Adapters*. |
| Cryptographic Message Syntax | Drop-down menu containing a list of cryptographic message syntaxes for building cryptographic messages. Required.Valid values are:<br>• CMS (default)<br>• PKCS#7 |

| Field | Description |
|---|---|
| Security Type | Drop-down menu containing the security type for building cryptographic messages. Required. Valid values are:<br><br>• Encrypted Only (default) - Encrypts the message only<br><br>• Detached Signed Only - Signs the original document and leaves the signature detached from the original document. If the message output format is SMIME, multipart MIME message will separate the original document and signature. If the message output format is DER or PEM, only detached signature will be returned by the service.<br><br>• Embedded Signed Only - Signs the original document and embeds the original document inside the signature.<br><br>• Detached Signed and Encrypted - Creates detached signed signature and encrypts the signed message. If the message output format is SMIME, the encryption is applied on the multipart MIME message. If the message output format is DER or PEM, the encryption is applied on the detached signature only.<br><br>• Embedded Signed and Encrypted - Creates embedded signed signature and encrypts the signed message. |
| Message Output Format | Message output format for generating the signed or encrypted message. Required. Valid values are:<br><br>• SMIME (default) - The signed or encrypted message will be output in MIME format.<br><br>• DER - The signed or encrypted message will be output in DER encoded format.<br><br>• PEM - The signed or encrypted message will be output in PEM encoded format, which is a base64 encoded DER format and enclosed between a start and an end boundary. |
| Document MIME Content Type | This parameter is enabled only if you select SMIME as the message output format. MIME content type for the document that needs to be packaged. If the input document is set with the content type, the value will override the setting here. Optional. Valid values are:<br><br>• application (default)<br><br>• text<br><br>• message<br><br>• image<br><br>• video<br><br>• audio |

| Field | Description |
|---|---|
| Document MIME Sub Content Type | This parameter is enabled only if you select SMIME as the message output formatMIME sub content type for the document that needs to be packages. If the input document is set with the sub content type, the value will override the setting here. Optional.Valid values are:<br><br>• octet-stream (default)<br>• plain<br>• edi-x12<br>• edifact<br>• edi-consent<br>• xml |
| Content Transfer Encoding | This parameter is enabled only if you select SMIME as the message output format.Content transfer encoding format. Optional.Valid values are:<br><br>• Base64 (default)<br>• None |
| Apply Content Transfer Encoding on Detached Document | This parameter is enabled only if you select SMIME as the message output format.To indicate if content transfer encoding should be applied on the detached document. This is used for *Detached Signed Only* and *Detached Signed and Encrypted* security types. Optional.Valid values are:<br><br>• Yes (default)<br>• No |
| Encryption Algorithm: | Content encryption algorithm. Optional.Valid values are:<br><br>• Triple DES (3DES) 168 CBC with PKCS5 padding (default)<br>• 56-bit DES CBC with PKCS5 padding<br>• 128-bit RC2 CBC with PKCS5 padding<br>• 40-bit RC2 CBC with PKCS5 padding<br>• 128-bit AES CBC with PKCS5 padding<br>• 192-bit AES CBC with PKCS5 padding<br>• 256-bit AES CBC with PKCS5 padding |
| Encryption Certificate(s): | Public certificates to encrypt the document. A list or a single certificate can be chosen to encrypt the same document. When you choose multiple certificates, it allows multiple recipients to decrypt the message. Optional. |
| Signature Options | Options to sign the message. Required.Valid values are:<br><br>• Single Signature (default)<br>• Multiple Signatures<br>• Counter Signature<br>• No Signature Required |
| Signing Algorithm | The signing algorithm to hash the document. Optional.Valid values are:<br><br>• SHA1 (default)<br>• MD5 |

| Field | Description |
|---|---|
| Signing Certificate(s) | Private certificates to sign the document. Optional.Valid values are:<br><br>• Select a signing certificate if you have selected Single Signature.<br><br>• Select a list of signing certificates for multiple users to sign the document if you have selected Multiple Signatures.<br><br>• Select a list of signing certificates for multiple users to sign the document and countersign the signature if you have selected Counter Signature. |
| Message Input Format | Message input format for parsing the signed or encrypted message. Required.Valid values are:<br><br>• SMIME (default)<br><br>• DER<br><br>• PEM |
| Security Type | This parameter is enabled only if you select either PEM or DER as the message input format.Security type that is applied to the inbound cryptographic message. Optional.Valid values are:<br><br>• Encrypted Only (default) - The inbound message is encrypted only.<br><br>• Detached Signed Only - The inbound message is signed in detached format.<br><br>• Embedded Signed Only - The inbound message is signed in embedded format.<br><br>• Detached Signed and Encrypted - The inbound message is signed in detached format and then encrypted.<br><br>• Embedded Signed and Encrypted - The inbound message is signed in embedded format and then encrypted. |
| Decryption Certificate | Private certificate to decrypt the cryptographic message. Optional. |
| Signature Verification Certificate(s) | Public certificates to verify signed cryptographic message. Optional.<br>**Note:** You can select single certificate if the inbound message is signed by one certificate or select a list of certificates if multiple certificates sign the inbound message. Based on the certificates list sequence, counter signature verification starts from the first level of the signature. |

## Parameters That Must Be Added in BPML

The following additional parameters are available for use with the Cryptographic Message service, but can only be added by editing your business process manually. This parameter is not available through the Admin console or the GPM:

| Parameter | Description |
|---|---|
| Action | The two values are either build or parse. Required. |

| Parameter | Description |
|---|---|
| pipelineTimeout | Controls the duration of building or parsing process. By default, the value is 300 seconds and can be increased to process large files. Optional. |

## Business Process Examples

The parameters passed from the BPML precede over the parameters passed from the service. The following BPML examples illustrate using the cryptographic message service instance:

**Example Business Process 1**

The following BPML builds the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```
<process name="cryptomsg_build">
  <sequence>
      <operation name="Crypto Message Service">
          <participant name="CryptoMsgService"/>
          <output message="buildRequest">
              <assign to="." from="*"/>
              <assign to="action">build</assign>
          </output>
          <input message="buildResponse">
              <assign to="." from="*"/>
          </input>
      </operation>
   </sequence>
</process>
```

**Example Business Process 2**

The following BPML parses the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```
<process name="cryptomsg_parse">
   <sequence>
      <operation name="Crypto Message Service">
          <participant name="CryptoMsgService"/>
          <output message="parseRequest">
              <assign to="." from="*"/>
              <assign to="action">parse</assign>
          </output>
           <input message="parseResponse">
               <assign to="." from="*"/>
           </input>
      </operation>
    </sequence>
</process>
```

**Example Business Process 3**

The following BPML builds and parses the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```
<process name="cryptomsg_buildandparse">
   <sequence>
    <operation name="Crypto Message Service">
```

```
            <participant name="CryptoMsgService"/>
             <output message="buildRequest">
               <assign to="." from="*"></assign>
               <assign to="action">build</assign>
               <!-- securityType=3 Encrypted Only,
                    securityType=1 Detached Signed Only,
                    securityType=2 Embedded Signed Only,
                     securityType=4 Detached Signed and Encrypted,
                      securityType=5 Embedded Signed and Encrypted -->
               <assign to="securityType">4</assign>
               <!-- signOptions=0 No Signature Required,
                    signOptions=1 Single Signature,
                    signOptions=2 Multiple Signatures,
                    signOptions=3 Counter Signature -->
               <assign to="signOptions">3</assign>
               <assign to="signAlgo">SHA1</assign>
               <assign to="signCerts">smime_priv1,smime_priv2,smime_priv3
</assign>
               <!-- encryption algorithm
             encAlgo=0 Triple DES 168 CBC with PKCS5 padding
                    encAlgo=1 56-bit DES CBCwith PKCS5 padding
                     encAlgo=2 128-bit RC2 CBC with PKCS5 padding
                     encAlgo=4 40-bit RC2 CBC with PKCS5 padding
                     encAlgo=6 128-bit AES CBC with PKCS5 padding
          encAlgo=7 192-bit AES CBC with PKCS5 padding
                    encAlgo=8 256-bit AES CBC with PKCS5 padding -->
               <assign to="encAlgo">0</assign>
               <assign to="encCerts">smime_pub1,smime_pub2</assign>
             </output>
              <input message="buildResponse">
                 <assign to="." from="*"></assign>
             </input>
           </operation>

         <operation name="Crypto Message Service">
            <participant name="CryptoMsgService"/>
            <output message="parseRequest">
               <assign to="." from="*"/>
                <assign to="action">parse</assign>
               <assign to="verifyCerts">smime_pub3,smime_pub2,smime_pub1
</assign>
     <assign to="decryptCert">smime_priv1</assign>
           </output>
            <input message="parseResponse">
               <assign to="." from="*"/>
           </input>
        </operation>
   </sequence>
</process>
```

**Example Business Process 4**

The following BPML puts the detached document under the *detachedDoc* area when parsing detaching only inbound message in PEM or DER format.

```
<process name="cryptomsg_parse">
   <sequence>
        <operation name="Import Document Request">
           <participant name="CryptoMsgTestFSA"/>
           <output message="FileSystemInputMessage">
               <assign to="Action">FS_COLLECT</assign>
               <assign to="collectionFolder" from="'/gisinstall'"/>
               <assign to="filter" from="'detached_doc.txt'"/>
                <assign to="useSubFolders">fals</assign>e
<assign to="useSubFolders">false</assign>
               <assign to="bootstrap">false</assign>
                <assign to="deleteAfterCollect">fals</assign>e
```

```
<assign to="deleteAfterCollect">false
            </assign>
         <assign to="." from="*"/>
     </output>
      <input message="FileSystemOutputMessage">
         <assign to="." from="*"/>
     </input>
   </operation>
    <assign to="detachedDoc" from="PrimaryDocument/@SCIObjectID"/>
   <operation name="Import Document Request">
      <participant name="CryptoMsgTestFSA"/>
      <output message="FileSystemInputMessage">
         <assign to="Action">FS_COLLECT</assign>
          <assign to="collectionFolder" from="'/gisinstall'"/>
         <assign to="collectionFolder" from="'/gisinstall'"/>
         <assign to="filter" from="'signed_msg.txt'"/>
 <assign to="useSubFolders">false</assign>
         <assign to="bootstrap">false</assign>
          <assign to="deleteAfterCollect">false</assign>
          <assign to="deleteAfterCollect">false</assign>
         <assign to="." from="*"/>
     </output>
      <input message="FileSystemOutputMessage">
         <assign to="." from="*"/>
     </input>
   </operation>
  <operation name="Crypto Message Service">
     <participant name="CryptoMsgService"/>
     <output message="parseRequest">
        <assign to="." from="*"/>
        <assign to="action">parse</assign>
        <!--securityType=3 Encrypted Only,
securityType=1 Detached Signed Only,
securityType=2 Embedded Signed Only,
     securityType=4 Detached Signed and Encrypted,
        securityType=5 Embedded Signed and Encrypted -->
        <assign to="securityType">1</assign>
        <!--msgFormat=0 SMIME,
           msgFormat=1 DER,
           msgFormat=2 PEM -->
        <assign to="msgFormat">2</assign>
        <assign to="verifyCerts">smime_pub1</assign>
     </output>
    <input message="parseResponse">
        <assign to="." from="*"/>
    </input>
  </operation>
 </sequence>
</process>
```

## Output from Service to Business Process

The following table describes the output from the cryptographic message service to the BPML ProcessData, when the service action is "build":

| Scenario | Output |
|---|---|
| Certificates used for encryption are acceptable | ```<br><EncryptCerts><br>    <Cert1><br>        <Name>smime_pub1</Name><br>        <Status>ok</Status><br>        <ExpiryTime>20350726074016Z</ExpiryTime><br>    </Cert1><br>    <Cert2><br>        <Name>smime_pub2</Name><br>        <Status>ok</Status><br>        <ExpiryTime>20350726074056Z</ExpiryTime><br>    </Cert2><br></EncryptCerts><br>``` |
| Certificate used for encryption or signing has expired | ```<br> <SigningCerts><br>    <Cert1><br>        <Name>smime_pub1</Name><br>        <Status>expired</Status><br>        <ExpiryTime>20070726074016Z</ExpiryTime><br>    </Cert1><br>  </SigningCerts><br><exception-message>xxx</exception-message><br>```<br><br>Or<br><br>```<br><EncryptCerts><br>    <Cert1><br>        <Name>smime_pub1</Name><br>     <Status>expired</Status><br>        <ExpiryTime>20070726074016Z</ExpiryTime><br>    </Cert1><br>     <Cert2><br>      <Name>smime_pub2</Name><br>        <Status>ok</Status><br>        <ExpiryTime>20350726074056Z</ExpiryTime><br>    </Cert2><br></EncryptCerts><br>``` |
| Certificate used for encryption has been revoked | ```<br><EncryptCerts><br>    <Cert1><br>        <Name>cert1</Name><br>        <Status>revoked</Status><br>    </Cert1><br></EncryptCerts><br><exception-message>xxx</exception-message><br>``` |
| Certificate used for encryption fails to process. For example, if the encryption certificate is not found in Sterling Integrator. | ```<br><EncryptCerts><br>    <Cert1><br>        <Name>cert1</Name><br>        <Status>error</Status><br>    </Cert1><br></EncryptCerts><br><exception-message>xxx</exception-message><br>``` |

The following table describes the output from the cryptographic message service to BPML ProcessData, when the service action is "parse":

| Scenario | Output |
|---|---|
| Decryption is passed | ```<br><DecryptionResult><br>    <DecryptionCertName>smime_priv1</DecryptionCertName><br>    <DecryptionCertStatus>ok</DecryptionCertStatus><br>    <DecryptionCertExpiryTime>20350726074016Z<br>    </DecryptionCertExpiryTime><br>    <Status>passed</Status><br></DecryptionResult><br>``` |

| Scenario | Output |
|---|---|
| Decryption certificate not found in Sterling Integrator | `<DecryptionResult>`<br>`    <DecryptionCertName>cert1</DecryptionCertName>`<br>`    <DecryptionCertStatus>error</DecryptionCertStatus>`<br>`    <Status>failed</Status>`<br>`</DecryptionResult>` |
| Decryption certificate failed to decrypt | `<DecryptionResult>`<br>`    <DecryptionCertName>smime_priv2</DecryptionCertName>`<br>`    <DecryptionCertStatus>ok</DecryptionCertStatus>`<br>`    <DecryptionCertExpiryTime>20350726074056Z`<br>`    </DecryptionCertExpiryTime>`<br>`    <Status>failed</Status>`<br>`</DecryptionResult>` |
| Signature verification is passed | `<SignatureVerificationResults>`<br>`    <SignatureVerificationResult1>`<br>`      <VerificationCertName>smime_dsa_pub`<br>`</VerificationCertName>`<br>`      <VerificationCertStatus>ok</VerificationCertStatus>`<br>`      <VerificationCertExpiryTime>20350812084354Z`<br>` </VerificationCertExpiryTime>`<br>`      <SigningTime>20080917021420Z</SigningTime>`<br>`      <Status>passed</Status>`<br>`    </SignatureVerificationResult1>`<br>`    <SignatureVerificationResult2>`<br>`      <VerificationCertName>smime_pub4</VerificationCertName>`<br>`      <VerificationCertStatus>ok</VerificationCertStatus>`<br>`      <VerificationCertExpiryTime>20350726074148Z`<br>` </VerificationCertExpiryTime>`<br>`      <SigningTime>20080917021420Z</SigningTime>`<br>`      <Status>passed</Status>`<br>`    </SignatureVerificationResult2>`<br>`    <Status>passed</Status>`<br>`</SignatureVerificationResults>` |
| Signature verification fails | `<SignatureVerificationResults>`<br>`    <SignatureVerificationResult1>`<br>`      <VerificationCertName>smime_pub4</VerificationCertName>`<br>`      <VerificationCertStatus>ok</VerificationCertStatus>`<br>`      <VerificationCertExpiryTime>20350726074148Z`<br>`</VerificationCertExpiryTime>`<br>`      <SigningTime>20080917021549Z</SigningTime>`<br>`      <Status>passed</Status>`<br>`    </SignatureVerificationResult1>`<br>`    <SignatureVerificationResult2>`<br>`      <VerificationCertName>smime_pub3</VerificationCertName>`<br>`      <VerificationCertStatus>ok</VerificationCertStatus>`<br>`      <VerificationCertExpiryTime>20350726074122Z`<br>`</VerificationCertExpiryTime>`<br>`      <SigningTime>20080917021549Z</SigningTime>`<br>`      <Status>failed</Status>`<br>`    </SignatureVerificationResult2>`<br>`    <Status>failed</Status>`<br>`</SignatureVerificationResults>` |

| Scenario | Output |
|---|---|
| Multiple signature verification fails | ```<br><SignatureVerificationResults><br>    <SignatureVerificationResult1><br>        <SigningTime>20080917071327Z</SigningTime><br>        <Status>nomatched_verificationCert</Status><br>    </SignatureVerificationResult1><br>    <SignatureVerificationResult2><br>        <VerificationCertName>smime_pub3</VerificationCertName><br>        <VerificationCertStatus>ok</VerificationCertStatus><br>        <VerificationCertExpiryTime>20350726074122Z<br></VerificationCertExpiryTime><br>        <SigningTime>20080917021549Z</SigningTime><br>        <Status>failed</Status><br>    </SignatureVerificationResult2><br>    <Status>failed</Status><br></SignatureVerificationResults><br>``` |
| Signature verification certificate is revoked | ```<br><SignatureVerificationResults><br>    <SignatureVerificationResult1><br>        <SigningTime>20080917024531Z</SigningTime><br>        <VerificationCertName>serenaCRL1</VerificationCertName><br>        <VerificationCertStatus>revoked</VerificationCertStatus><br>    </SignatureVerificationResult1><br>    <Status>failed</Status><br></SignatureVerificationResults><br>``` |

The CMS service allows you to use an expired certificate to encrypt/decrypt or sign/verify the message if "validity" flag is not enabled when you check in the certificate into the system. The certificate status and expiry time is shown in the ProcessData as part of CMS service output.

The certificate ExpiryTime and SigningTime is displayed in UTC timezone in yyyyMMddHHmmssZ format. The BPML can perform the following checks after calling the CMS service:

- ExpiryTime against SigningTime to determine if the signature verified by the expired certificate is acceptable or not.
- ExpiryTime against the current date to determine if the encrypted or signed data created the expired certificate is acceptable or not.

# Chapter 49. Data Sweeper Service (5101 and higher)

The following table provides an overview of the Data Sweeper service:

| System name | Data Sweeper Service |
|---|---|
| Graphical Process Modeler (GPM) categories | System Services |
| Description | The Data Sweeper service is an optionally scheduled system service that cleans up data that is not in use and not cleaned by other system clean up processes due to lack of any continued associations to the data. |
| Business usage | The Data Sweeper service is a system service that corrects discovered entity relationship issues within the database that could potentially cause performance and unnecessary database expansion. |
| Usage examples | Based on the optional parameters and dataSweeper.properties file settings, you have set, Data Sweeper service cleans up the data potentially left from data disassociations from the following tables:<br>• EDIINT<br>• Correlations<br>• Document Clones<br>• Document Life Span<br>• GUID<br>• Performance Engine Stats<br>• Workflow Context<br>• Workflow ID<br><br>**Note:** The Data Sweeper command line option (`datasweeper.cmd` or `datasweeper.sh`) also cleans Data_Table/TRANS_DATA on the recommendation of the Sterling Customer Support. |
| Preconfigured? | Yes. DataSweeper. |
| Requires third party files? | No |
| Platform availability | The following platforms are supported:<br>• HP-UX<br>• IBM AIX<br>• IBM iSeries (OS/400)<br>• Microsoft Windows 2000<br>• RedHat AS<br>• Sun Solaris<br>• United Linux |
| Related services | None<br>**Note:** The Data Sweeper service references the `dataSweeper.properties` file in the `<SIInstallDir>/properties` directory. |

| System name | Data Sweeper Service |
|---|---|
| Application requirements | None |
| Initiates business processes? | Data Sweeper service is a system service that runs a business process. You can run `DataSweeper.sh` or `DataSweeper.cmd` from the command line. |
| Invocation | The Data Sweeper service is not for use in customer business processes but you may use it in a system business process. |
| Business process context considerations | None |
| Returned status values | • Success<br>• Failure |
| Restrictions | To run -dataTableScanSweeper command line option, ensure that the Application is shut down, and the database is running. Use the dataTableScanSweeper option only if Customer Support advises you to run it. **Note:** Data Sweeper service starts MySQL if it is already not running. |
| Persistence level | The default persistence level is Full. You can set the persistence to a lower level if logging is not required. |
| Testing considerations | Data Sweeper service writes to the noapp log file. |

## Configuring the Data Sweeper Service

There are no configurable parameters. All options must be set in the BPML, at the command line, or in the properties file. At run time, the command line or BPML will override the properties file settings in the case of a conflict.

## Business Process Example

The following example business process illustrates using the Data Sweeper service:

```
<process name="Schedule_DataSweeper">
    <rule name="obtainLock">
        <condition>controlLock ='true' </condition>
    </rule>
    <sequence>
    <assign to='controlLock'>false</assign>
    <operation name="SetLock">
        <participant name="SystemLockService"/>
        <output message="Xout">
          <assign to="LOCK_KEY">DataSweeper</assign>
          <assign to="DURATION">86400000</assign>
    <assign to="CLEAR_ON_START_UP">true</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="Xin">
          <assign to="." from="*"></assign>
        </input>
      </operation>
     <assign to='controlLock'>true</assign>
      <operation name="Service">
        <participant name="DataSweeper"/>
        <output message="Xout">
```

```
                  <assign to="batchSize">5000</assign>
                  <assign to="autocorrect">TRUE</assign>
                  <assign to="maxIterations">1000</assign>
                  <assign to="sweeperTimeout">1080000</assign>
                  <assign to="sweeperTimeoutThreshold">36000000</assign>
                  <assign to="." from="*"></assign>
               </output>
               <input message="Xin">
                  <assign to="." from="*"></assign>
               </input>
            </operation>
         <operation name="unLock">
           <participant name="SystemLockService"/>
           <output message="Xout">
             <assign to="ACTION">unlock</assign>
             <assign to="LOCK_KEY">DataSweeper</assign>
             <assign to="." from="*"></assign>
           </output>
           <input message="Xin">
             <assign to="." from="*"></assign>
           </input>
          </operation>
        <onFault>
          <sequence name="LockFaild">
            <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">DataSweeper</assign>
                    <assign to="." from="*"></assign>
                  </output>
                  <input message="Xin">
                    <assign to="." from="*"></assign>
                  </input>
                </operation>
              </sequence>
              <sequence name="stopWithoutLocking">
                <assign to="Document/Msg" append="true">Failed to obtain a lock.!</assign>
              </sequence>
            </choice>
            <assign to="Document/Status" append="true">Failed!</assign>
            <assign to="Document/Msg" append="true">DataSweeper failed!</assign>
            <assign to="Document/CurrentHost" append="true">loki</assign>
            <assign to="Document/CurrentPort" append="true">53000</assign>
            <assign to="Document/DetailMsg" from="/ProcessData/StatusRpt/text()"
append="true"></assign>
          <operation name="SMTP Send">
              <participant name="SMTP_SEND_ADAPTER"/>
              <output message="SMTP_SEND_ADAPTERInputMessage">
                <assign
to="xport-smtp-mailfrom">alert_email_recipient@yournet.com</assign>
                <assign to="xport-smtp-mailhost">yourmailhost.local</assign>
                <assign to="xport-smtp-mailport">25</assign>
                <assign to="xport-smtp-mailto">alert_email_recipient@yournet.com</assign>
                <assign to="xport-smtp-mailsubject">Automated Event Notification -
DataSweeper Failed</assign>
                <assign to="PrimaryDocument" from="DOMToDoc(Document)/@*"></assign>
                <assign to="." from="*"></assign>
              </output>
            <input message="inmsg">
              <assign to="." from="*"></assign>
```

```
          </input>
        </operation>
     </sequence>
  </onFault>
   </sequence>
  </process>
```

# Chapter 50. DB Monitor Service

The DB Monitor service monitors the database for common conditions that may be signs of problems with database health. The following table provides an overview of the DB Monitor service:

| System Name | DBMonitorService |
|---|---|
| Graphical Process Modeler (GPM) categories) | All Services |
| Description | This service monitors the database for common conditions that may be signs of system problems. May be scheduled to run at regular intervals. |
| Business usage | The DB Monitor service is used by the Schedule_DBMonitorService business process. Together they are used by the system to notify the system administrator if key database health thresholds are exceeded or if maintenance is needed. This maintenance includes indexes, rebuilds, and analyze table actions. By default, this service is scheduled to run every Monday at 4:00 A.M. |
| Usage example | The DB Monitor service is scheduled to run once a week. If the service determines that a key indicator has exceeded its threshold, the system will notify the system administrator by e-mail. It also monitors the database maintenance needs and provides the instructions (SQL commands) necessary to rebuild indexes and analyze tables. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | None |
| Application requirements | The database must support the requested functionality. Functionality available is dependent on the database's ability to support it. For example, DB2 and MySQL do not support providing statistics on database size. |
| Initiates business processes? | The DB Monitor service invokes an associated system business process, Schedule_DBMonitorService. |
| Invocation | Scheduler |
| Business process context considerations | None |
| Returned status values | N/A |
| Restrictions | None |
| Persistence level | System Default |
| Testing considerations | None |

## How the DB Monitor Service Works

The DB Monitor service sends notification, as configured by the Sterling Integrator event alert system, of any recommended maintenance. If the DB Monitor service determines that the system administrator needs to rebuild the index, or analyze database tables, the service will produce a primary document that provides

**299**

information about the recommended maintenance. The DB Monitor service will also provide notification if key database health thresholds are exceeded. By default, the system will send an e-mail to the system administrator.

The preconfigured instance of the DB Monitor service, named DBMonitorService, has an associated business process named Schedule_DBMonitorService that performs all the available actions of the DB Monitor service using default values for each. The DBMonitorService instance is scheduled to run each Monday at 4:00 A.M. You can change the run day and time using the Scheduler.

If you create a new instance of the DB Monitor service and specify a schedule in the service configuration, the system will create an associated business process using the name of your instance (minus any spaces) preceded by `Schedule_`. For example, if you name your instance *My DB Monitor Service*, the business process name will be `Schedule_MyDBMonitorService`. If you do not use a schedule in the service configuration, a business process will not automatically be created.

You can change settings by editing the BPML in the associated business process. You can also use the DB Monitor service in your own business process and specify settings in the Graphical Process Monitor, or directly in your BPML.

## Implementing the DB Monitor Service

To implement the DB Monitor service, complete the following tasks:

1. Create a new configuration of the DB Monitor service. For basic information about creating service configurations, see *Managing Services and Adapters*.
2. Specify field settings for scheduling the service configuration in the Sterling Integrator Admin Console.
3. If necessary, specify settings in the GPM, or using BPML in your business processes. This step is not necessary if you want to use the pre-configured business process with default settings.

## Configuring the DB Monitor Service

## Creating a Service Configuration in the Admin Console

Use the field definitions in the following table to create a new configuration of the DB Monitor service.

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

| Field | Description |
|---|---|
| Select a Group | Select one of the options: |
| | • None – Do not include the configuration in a service group at this time. |
| | • Create New Group – Enter a unique name for a new group, which will be created with this configuration. (You can then add other services to the group as well.) |
| | • Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |
| | **Note:** For more information about service groups, see *Managing Services and Adapters*. |
| Run As User | Type the user ID to associate with the schedule, or click the |
| | 🖼️ |
| | icon and select a user ID from the list. Valid values: Any valid Sterling Integrator user ID. |
| Use 24 Hour Clock Display | Check the box to use the 24-hour clock instead of the default 12-hour clock. |
| Schedule | Specify the scheduling information for running the Auto Terminate service. Valid values: |
| | • Do not use schedule If you select this field, the service does not run on a schedule. |
| | • Run based on timer Valid values are the hour and minutes at which to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the service to run at startup. |
| | • Run daily Valid values are the hour and minutes at which to run the service each day. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the service to run at startup. |
| | • Run based on day(s) of the week Valid values are the day of the week, the hour, and the minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |
| | • Run based on day(s) of the month Valid values are the day of the month, hour, and minute that specify when to run the service. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |
| | **Note:** Any value except **Do not use schedule** will cause a Schedule_*InstanceName* business process to be automatically created. See *How the DB Monitor Service Works* for more information. If you select **Do not use schedule**, you must specify service settings 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. |
| ACTION | Specifies an action to take. Required. Valid Values:<br>• Check DB Full – Check the current size of the database and provide the result as a percentage of the maximum database size.<br>• Check DB Grown – Check how much the database has grown since the last time it was checked and provide the result as a percentage.<br>• Check DB Index – Check whether indexes need to be rebuilt or check whether tables need to be analyzed. Either **REBUILD_INDEXES** or **ANALYZE** must be set to true.<br><br>**Note:** The Schedule_*instanceName* business process that is automatically created with an implementation of the DB Monitor service performs all three actions using the default values for each. |
| ANALYZE | Used only if **ACTION** is set to **Check DB Index**. Specifies whether to check whether tables need to be analyzed. Optional. Valid values:<br>• true – Check whether tables need to be analyzed.<br>• false – Do not check whether tables need to be analyzed. Default. |
| DO_AII_TABLE | Used only if **ACTION** is set to **Check DB Index**. Specifies whether to check all the tables from the database. Optional. Valid values:<br>• true – Check all the database tables.<br>• false – Check the tables that recorded on the database since last time rebuilds Default. |
| LAST_ANALYZED_DATE | Used only if **ACTION** is set to **Check DB Index**. The expiration date and time to check against the database. If the table is expired since the last check, it will placed in a list for index rebuilding or analysis. Optional. Example value: 20051230-12:12:12.000. Default is current date. |
| MAX_PERCENT_FULL | Used only if **ACTION** is set to **Check DB Full**. The percentage of maximum database size the database must reach before a notification is sent to the event framework. Optional. Default is 80. |
| MAX_PERCENT_GROWN | Used only if **ACTION** is set to **Check DB Grown**. The percentage of maximum database size that the database must have grown since last time checked before a notification is sent to the event framework. Optional. Default is 30. |
| REBUILD_INDEXES | Used only if **ACTION** is set to **Check DB Index**. Specifies whether to check whether the indexes need rebuilt. Optional. Valid values:<br>• true – Check whether indexes need to be rebuilt.<br>• false – Do not check whether indexes need to be rebuilt. Default. |

| Field | Description |
|-------|-------------|
| SAVE_STATUS | Used only if **ACTION** is set to **For Check DB Full** or **Check DB Grown**. Specifies whether to save the database usage to the database to allow comparison of the difference for the next check. Optional. Valid values<br><br>• true – Save the database usage to the database.<br><br>• false – Do not save the database usage to the database. Default. |

## Business Process Example

The following example illustrates a business process containing the DB Monitor service. The following BPML is from the Schedule_DBMonitorService business process invoked by the DB Monitor service.

```
<process name="Schedule_DBMonitorService">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="SystemLockService"/>
      <output message="Xout">
        <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="CLEAR_ON_START_UP">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
Check for database percentage grown beyond threshold since last check
    <sequence name="Process">
      <operation name="DBMonitorService_1">
        <participant name="DBMonitorService"/>
        <output message="Xout">
          <assign to="ACTION">CHECK_DB_GROWN</assign>
          <assign to="MAX_PERCENT_GROWN">30</assign>
          <assign to="SAVE_STATUS">true</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="Xin">
          <assign to="." from="*"></assign>
        </input>
      </operation>
```

Check for database percentage full.

```
      <operation name="DBMonitorService_2">
        <participant name="DBMonitorService"/>
        <output message="Xout">
          <assign to="ACTION">CHECK_DB_FULL</assign>
          <assign to="MAX_PERCENT_FULL">80</assign>
          <assign to="SAVE_STATUS">true</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="Xin">
          <assign to="." from="*"></assign>
        </input>
      </operation>
Check for whether or not the database should be re-indexed
      <operation name="DBMonitorService_3">
        <participant name="DBMonitorService"/>
        <output message="Xout">
          <assign to="ACTION">CHECK_DB_INDEX</assign>
```

```
                              <assign to="REBUILD_INDEXES">true</assign>
                              <assign to="ANALYZE">true</assign>
                              <assign to="DO_All_TABLE">false</assign>
                              <assign to="." from="*"></assign>
                        </output>
                        <input message="Xin">
                              <assign to="." from="*"></assign>
                        </input>
                  </operation>
                  <sequence name="UnLock">
                        <operation name="UnLock">
                              <participant name="SystemLockService"/>
                              <output message="Xout">
                                    <assign to="ACTION">unlock</assign>
                                    <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
                                    <assign to="." from="*"></assign>
                              </output>
                              <input message="Xin">
                                    <assign to="." from="*"></assign>
                              </input>
                        </operation>
                        <onFault>
                              <assign to="UnLock_Msg" append="true">Failed to obtain a unlock!
</assign>
                        </onFault>
                  </sequence>
                  <onFault>
                        <operation>
                              <participant name="SystemLockService"/>
                              <output message="Xout">
                                    <assign to="ACTION">unlock</assign>
                                    <assign to="LOCK_KEY">DBMonitorService_Lock</assign>
                                    <assign to="." from="*"></assign>
                              </output>
                              <input message="Xin">
                                    <assign to="." from="*"></assign>
                              </input>
                        </operation>
                  </onFault>
            </sequence>
            <onFault>
                  <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
            </onFault>
      </sequence>
</process>
```

## Parameters Passed From Service to Business Process

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

| Parameter | Description |
|---|---|
| UsagePercentFull | The current percent used of maximum database size. |
| UsageUsed | The current amount of space, in MB, used by the database. |
| UsageTotal | The total amount of space, in MB, used by the database. |
| UsageUsedKB | The current amount of space, in KB, used by the database. |
| PercentGrown | The percentage the database has grown since the last time it was checked. |
| SQL_<Number_of_SQL> | The actual SQL statements for index rebuilds and analyze tables. |

# Chapter 51. Document Keyword Replace Service

The Document Keyword Replace service enables you to find and replace text within a document.

The following table provides an overview of the Document Keyword Replace service:

| System name | Document Keyword Replace Service |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | The Document Keyword Replace Service allows you to replace user-defined keywords in a document. This can be used to manipulate hexadecimal or string values in a document. A keyword can be replaced with either a replacement string or the contents of a document. |
| Preconfigured? | No |
| Business usage | Because of disparities between how different trading partners' enterprise systems process data, the format of the data that is exchanged between trading partners often needs to be manipulated before sending or after receiving or in some cases, on both sides of the transmission. |
| Usage example | Assume that the VAN used by your trading partner expects each line of data to be terminated with a Carriage Return and Line Feed, and the data from your enterprise system is terminated by tilde (~) characters. You would use the DocKeywordReplace service to convert the tildes into Carriage Return / Line Feed characters, before transmitting the data to your trading partner. |
| Requires third party files? | No |
| Platform availability | All supported 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 | This service uses the primary document. |
| Returned status values | Returned status values:<br>• Success – Document Keyword Replace service was successful.<br>• Error – Document Keyword Replace service was unsuccessful. |
| Restrictions | None |

## Implementing the Document Keyword Replace Service

To implement the Document Keyword Replace service, complete the following tasks:

1. Create a Document Keyword Replace service configuration. See *Creating a Service Configuration*.
2. Configure the Document Keyword Replace service. See *Configuring the Document Keyword Replace Service*.
3. Use the Document Keyword Replace service in a business process.

## Configuring the Document Keyword Replace Service

To configure the Document Keyword Replace service, use the following BPML parameters in your business process:

| BPML | Description |
|---|---|
| keyword<x> | Keyword that the service should look for. Required. <x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: **keyword1**, **keyword2**, **keyword3**, etc.<br>**Note:** Each **keyword** must have a corresponding **replace**. Otherwise, it will be ignored and you may receive an error message. All subsequent **keyword**<x>/**replace**<x> pairs will also be ignored.<br>**Note:** Always define each **keyword**<x>/**replace**<x> pair sequentially. For example, do not specify a **keyword2**/**replace2** pair and a **keyword4**/**replace4** pair without a **keyword3**/**replace3** pair between them. |
| keywordtype<x> | Type of keyword. Optional. Valid values are hex, string, and encoding type for the parameter keyword<x>. Encoding type is any supported encoding format such as UTF-8 and UTF-16. Default is string.<br>**Note:** Encoding is supported only when literal_mode is set to TRUE.The following example demonstrates the use of UTF-8 as the encoding type for the parameter keyword<x>:<br><pre>&lt;output message="DocKeywordReplaceInputMessage"&gt;<br>    &lt;assign to="literal_mode"&gt;true&lt;/assign&gt;<br>    &lt;assign to="keyword1" from "'&amp;#x6E;'"&gt;<br>&lt;/assign&gt;<br>    &lt;assign to="keywordype1"&gt;UTF-8&lt;/assign&gt;<br>    &lt;assign to="replace1" from="'&amp;#x7F;'"&gt;<br>&lt;/assign&gt;<br>    &lt;assign to="replacetype1"&gt;UTF-8&lt;/assign&gt;<br>&lt;/output&gt;</pre><br><x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: **keywordtype1**, **keywordtype2**, **keywordtype3**, etc.<br>**Note:** This parameter is optional in either mode.<br>**Note:** If double byte characters are specified in the keyword<x> parameter, this parameter must be set. Additionally, if double byte characters are specified in the keyword<x> or replace<x> parameters, this parameter must be set to string. |

| BPML | Description |
|------|-------------|
| replace<x> | Replacement text that the service should use to replace the keyword. Required. Valid values are hex and string. Default is string.<x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: **replace1**, **replace2**, **replace3**, etc.<br>**Note:** Each **replace** must have a corresponding **keyword**. Otherwise, it will be ignored and you will receive an error message.<br>**Note:** Always define each **keyword**<x>/**replace**<x> pair sequentially. For example, do not specify a **keyword2**/**replace2** pair and a **keyword4**/**replace4** pair without a **keyword3**/**replace3** pair between them. |
| replacetype<x> | Type of replacement text. Optional. Valid values are hex, string, and encoding type for the parameter replace<x>. Encoding type is any supported encoding format such as UTF-8 and UTF-16. Default is string.<br>**Note:** Encoding is supported only when literal_mode is set to TRUE.The following example demonstrates the use of UTF-8 as the encoding type for the parameter replace<x>:<br><pre><output message="DocKeywordReplaceInputMessage"><br>    <assign to="literal_mode">true</assign><br>    <assign to="keyword1" from="'&amp;#x6E;'"><br></assign><br>    <assign to="keywordype1">UTF-8</assign><br>    <assign to="replace1" from="'&amp;#x7F;'"><br></assign><br>    <assign to="replacetype1">UTF-8</assign><br><br></output></pre><x> refers to a sequential number starting with 1. This allows you to make multiple replacements. For example: **replacetype1**, **replacetype2**, **replacetype3**, etc.<br>**Note:** This parameter is optional in either mode.<br>**Note:** If double byte characters are specified in the replace<x> parameter, this parameter must be set. Additionally, if double byte characters are specified in the keyword<x> or replace<x> parameters, this parameter must be set to string. |
| keystart | Characters to add to the beginning of the keyword when the service looks for the string to replace. Optional. Default is ${.<br>**Note:** Do not specify a value for this parameter if **useKeywordDefaults** is set to true and you are using **String** mode.<br>**Note:** This parameter can also be specified in Sterling Integrator or in the GPM. However, special characters (for example, characters such as parentheses and brackets) *must* be specified using BPML.<br>**Note:** This parameter is not used if **literal_mode** is set to true. |

| BPML | Description |
| --- | --- |
| keyend | Characters to add to the end of the keyword when the service looks for the string to replace. Optional. Default is }.<br>**Note:** Do not specify a value for this parameter if `useKeywordDefaults` is set to true and you are using **String** mode.<br>**Note:** This parameter can also be specified in Sterling Integrator or in the GPM. However, special characters (for example, characters such as parentheses and brackets) *must* be specified using BPML.<br>**Note:** This parameter is not used if `literal_mode` is set to true. |
| mode | Specifies the mode to use. Optional. Valid values:<br>• Document – streams in a complete document in place of a keyword<br>• String – does a simple string replacement<br>Default is String.<br>**Note:** This parameter is not used if `literal_mode` is set to true. |
| useKeywordDefaults | Whether to add the characters specified in **keystart** and **keyend**, or their default values, to the start and end of the keyword you are looking for. Optional. Valid values:<br>• true – Add the characters specified in **keystart** and **keyend**, or the default values, to the start and end of the keyword. This is the default setting.<br>• false – Do not add any characters to the start and end of the keyword.<br>**Note:** This parameter is not used if `literal_mode` is set to true. |
| literal_bufferSize | Specifies the size of the buffer used for literal replacements. Optional. Default is 10240 bytes (10 KB).<br>**Note:** This parameter can also be specified in the Graphical Process Modeler (GPM). |
| literal_mode | Whether to use high speed literal replacements (literal mode). Valid values: true, false.<br>**Note:** Literal mode provides high speed performance by limiting processing overhead. You should use this mode unless you specifically need the capabilities of the other modes. The following parameters are not used if `literal_mode` is set to true:<br>• keystart<br>• keyend<br>• mode<br>• useKeywordDefaults<br>**Note:** This parameter can also be specified in the GPM. |
| literal_readAheadSize | Specifies the size of the read-ahead buffer used for literal replacements. Optional. Default is 8192 bytes (8 KB). To ensure proper operation, the value specified for this parameter must be smaller than the value specified for the `literal_bufferSize` parameter.<br>**Note:** This parameter can also be specified in the GPM. |

## Choosing a Mode

There are three modes available in the Document Keyword Replace service.

**Literal Mode**

Because Literal mode operation limits processing overhead and performs the search and replace functions only once, this mode is ideal for high-speed replacement operations. You should always use the Document Keyword Replace service in Literal mode, unless you:

- need to replace a keyword with the contents of a document (use Document mode).
- need recursive replacement capability (use String mode).
- are working with the Synchronization Engine adapters (see *Synchronization Engine Adapters*).
- need parameters not available in Literal mode (use String mode).

  **Note:** Because Literal mode limits processing overhead, some parameters are not available in Literal mode. Use String mode if you need these parameters. See *literal_mode* for a list of these parameters.

Place the service in Literal mode by setting the `literal_mode` parameter to true. For more information on setting this parameter, see *Configuring the Document Keyword Replace Service*.

In literal mode, you can also support character deletion.

**String Mode**

The String mode offers more options and the ability to do recursive replacements. This is the default mode. Place the Document Keyword Replace service in String mode by setting the `mode` parameter to String, or by not specifying a mode. Ensure the `literal_mode` parameter is set to false. For more information on setting this parameter, see *Configuring the Document Keyword Replace Service*.

String mode allows the use of the following parameters:
- keywordtype
- replacetype
- keystart
- keyend
- mode
- useKeywordDefaults

Some situations where you should use String mode include:
- You need the optional parameters available in String mode
- You are working with the Synchronization Engine adapters (see *Synchronization Engine Adapters*)
- You need to do recursive replacements

  **Note:** Because String mode operates recursively, the service contains a mechanism to prevent endless loops when using String mode. You should always test your configuration. If you find that any expected replacements were not made, check your `keyword` and `replace` parameter settings. For example, if

you set up the Document Keyword Replace service to replace the string "ABC" with the string "ABCD", while in String mode, the service will not allow it. Because the **replace** parameter value (ABCD) contains the **keyword** parameter value (ABC), if the service was allowed to run, the result would be an endless series of replacement iterations as shown below:

– Iteration 1 – ABC

– Iteration 2 – ABCD

– Iteration 3 – ABCDD

– Iteration 4 – ABCDDD

– Iteration 5 – ABCDDDD

– This would continue in the same manner endlessly

**Document Mode**

If you need to replace a string with a complete document, use the Document mode. Place the Document Keyword Replace service in Document mode by setting the **mode** parameter to Document. For more information on setting this parameter, see *Configuring the Document Keyword Replace Service*.

## Business Process Examples

In the following examples, a notation of #x?? is the hexadecimal equivalent of an unprintable character (?? represents a hexadecimal number).

**Note:** You cannot just cut and paste (and then run) these examples; you must insert the process name at the beginning of the BPML (such as the following example in which you would replace "somename" with the actual name of the process) and ensure you defined a valid service instance:

```
<process name="somename">
  <!-- within a process, you can have sequences, assigns, and operations-->
  </process>
```

**Literal Mode**

This sample Business Process searches for all line feed characters and replaces them with a carriage return and tilde:

**Note:** In the example below, &#x0a; is a hex notation of a "character entity reference" that is defined by html (see http://www.w3.org/TR/REC-html40/sgml/entities.html for more information).

```
<process name="test">
<sequence>
<operation name="DocKeywordReplaceImpl">
<participant name="myDocKeywordReplaceImplService"/>
<output message="outmsg">
<assign to="." from="*" />
<assign to="literal_mode">true</assign>
<assign to="literal_bufferSize">102400</assign>
<assign to="literal_readAheadSize">8192</assign>
<assign to="keyword1">~</assign>
<assign to="replace1" from="string('&#x0a;')"/>
<assign to="keyword2" from="string('&#x0d;&#x7e;')"/>
<assign to="replace2">Ninety</assign>
</output>
<input message="inmsg">
<assign to="." from="*" />
```

```
</input>
</operation>
</sequence>
</process>
```

For the sample business process above, this is the Primary Document before it is run through the Document Keyword Replace service:

```
ISA*00*  *00*  *08*9262390000  *14*9252050234*020129*0525*U*00400*000010006*0*P*:
GS*PD*345345345*9252050234*20020129*0525*10006*T*004010
ST*852*000010874
XQ*G*20020128*20020202
XPO*L774211-01
N9*VR*0103103*MEASLES ANIMAL HAVEN
N1*ST*CHART WHSE - LKLD - (942)*9*0069220090942
LIN**UA*003500005221
ZA*QA*594*CA*007*20020128
LIN**UA*003500048553
ZA*QA*594*CA*007*20020128
LIN**UA*003500005537
ZA*QA*576*CA*007*20020128
LIN**UA*003500048571
ZA*QA*800*CA*007*20020128
LIN**UA*003500048501
ZA*QA*600*CA*007*20020128
CTT*5
SE*1*000010874
GE*1*10006
IEA*1*000010006
```

For the sample business process above, this is the Primary Document after it is run through the Document Keyword Replace service:

```
ISA*00*  *00*  *08*9262390000  *14*9252050234*020129*0525*U*00400*000010006*0*P*:~
  ß----- Note a Tilde was added to the end of each line
GS*PD*345345345*9252050234*20020129*0525*10006*T*004010~
ST*852*000010874~
XQ*G*20020128*20020202~
XPO*L774211-01~
N9*VR*0103103*MEASLES ANIMAL HAVEN~
N1*ST*CHART WHSE - LKLD - (942)*9*0069220090942~
LIN**UA*003500005221~
ZA*QA*594*CA*007*20020128~
LIN**UA*003500048553~
ZA*QA*594*CA*007*20020128~
LIN**UA*003500005537~
ZA*QA*576*CA*007*20020128~
LIN**UA*003500048571~
ZA*QA*800*CA*007*20020128~
LIN**UA*003500048501~
ZA*QA*600*CA*007*20020128~
CTT*5~
SE*1*000010874~
GE*1*10006~
IEA*1*000010006~
```

**Document Mode**

**Note:** The path of the replace*X* parameter must include a document name from ProcessData. In the following example, `replacementDocument` is the name of the document in ProcessData.

```
<process name="test">
 <sequence>
  <operation name="DocKeywordReplace">
  <participant name="myDocKeywordReplaceService"/>
  <output message="toService">
    <assign to="." from="*"/>
    <assign to="mode">Document</assign>
    <assign to="keyword1">someKeywordToReplace</assign>
    <assign to="replace1">/ProcessData/replacementDocument</assign>
    <assign to="keyword2">anotherKeywordToReplace</assign>
  <assign to="replace2">/ProcessData/PrimaryDocument</assign>
```

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

## String Mode

```
<process name="test">
  <sequence>
    <operation name="DocKeywordReplaceImpl">
     <participant name="myDocKeywordReplaceImplService"/>
     <output message="outmsg">
       <assign to="." from="*" />
       <assign to="keyword1">GLN</assign>
       <assign to="replace1">0008586666221</assign>
       <assign to="keyword2">GTIN</assign>
       <assign to="replace2" from="string(myData/GTIN)" />
       <assign to="keyword3">USERNAME</assign>
       <assign to="replace3" from="'user1'" />
       <assign to="keystart" from="'$('" />
       <assign to="keyend" from="'$)'" />
     </output>
     <input message="inmsg">
       <assign to="." from="*" />
     </input>
   </operation>
  </sequence>
</process>
```

## Character Deletion Example

Sterling Integrator now supports character deletion as exemplified in the following BPML assignment:

```
<process name="DocKeywordRemoveExample">
  <sequence name="Sequence Start">
    <operation name="Document Keyword Replace">
      <participant name="SomeDocKeywordReplaceInstance"/>
      <output message="DocKeywordReplaceInputMessage">
        <assign to="." from="*"/>
        <assign to="literal_mode">true</assign>
        <assign to="literal_bufferSize">102400</assign>
        <assign to="literal_readAheadSize">8192</assign>
        <assign to="keyword1">111111</assign>
        <assign to="replace1" from="string('')"/>
        <assign to="keyword2">ABC</assign>
        <assign to="replace2" from="string('')"/>
      </output>
      <input message="inmsg">
        <assign to="." from="*"/>
      </input>
    </operation>
  </sequence>
</process>
```

# Chapter 52. Document Tracking Service

The following table provides an overview of the Document Tracking service:

| System name | DocumentTracking |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | Enables you to define correlation name-value pairs to gather additional tracking data about documents. |
| Business usage | Enables you to gather additional tracking data. You can include one or more tracking services at appropriate points in a business process and define additional tracking data (correlation name-value pairs) to be gathered about a document (or a family of documents). |
| Usage example | When setting up a business process, you can include one or more tracking services. While checking-in the business process, if document tracking has been enabled, the information discovery process will be responsible for creating the document tracking record (and populate it with an unique tracking ID along with pre-defined tracking data) and the tracking service will add to the contents of the tracking record. However, if document tracking has not been enabled, the tracking service will create the tracking record. When the business process is executed and documents are processed, tracking data about the documents is gathered and passed from one step in the process to the next. If, during one of the steps in the process, a document A gets split into 3 documents, A1, A2 and A3, all 4 documents would share a common tracking ID—the relationships between documents within a family are available when a user searches for documents using tracking information as the search criteria. |
| Preconfigured? | Yes (DocumentTracking) |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | None |
| Application requirements | None |
| Business process context considerations | None |
| Returned status values | Returned status values: <br> • Success–The service successfully created or updated a tracking record. <br> • Error–The service was not successful in creating or updating a tracking record. |
| Restrictions | Requires a primary document. |

## Implementing the Document Tracking Service

To implement the Document Tracking service, complete the following tasks:

1. Create a Document Tracking service configuration. See *Creating a Service Configuration*.

2. Configure the Document Tracking service. See *Configuring the Document Tracking Service*.

3. Use the Document Tracking service in a business process.

## Configuring the Document Tracking Service

To configure the Document Tracking service, you must specify field settings in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| use-system-default | The system default is to perform document tracking if it is enabled at the business process level. Optional. Valid values are:<br>• True – Tracking is performed if it is enabled at the business process level.<br>• False – Extent of the tracking operation depends on the discovery-level setting. Default. |
| discovery-level | Optional. Valid values are:<br>• None – Information discovery is not performed. Default.<br>• Optional – Information discovery is not performed, if it has not been performed previously.<br>• Mandatory – Information discovery is performed, even if it has been performed previously. |
| tracking-scope | Tracking scope is an advanced option that is used to update system correlations for better control over the system. Optional.<br>**Note:** Changing from the default scope (Custom) may cause conflicts with existing correlations. This custom parameter can be added manually through BPML or through the Advanced editor in the Graphical Process Modeler.Valid values are:<br>• Custom<br>• System<br>• Adapter<br>• EDI<br>• EDIInterchange<br>• EDIGroup<br>• EDITransaction<br>• ACH<br>• CD<br>• MQ<br>• Translation<br>• EDIINT<br>• Mailbox |

## Business Process Example

The following example illustrates how the Document Tracking service could be used in a business process:

```
<process name="TestDocTracking">
  <sequence>
    <!-- Using custom correlation -->
    <operation name="Tracking">
      <participant name="DocumentTracking"/>
      <output message="msg-to-service">
        <assign to="my-Correlation" from="'myValue'"/>
        <assign to="my-Other-Correlation" from="'myOtherValue'"/>
        <assign to="." from="PrimaryDocument"/>
      </output>
      <input message="msg-from-service">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <!-- Changing system state where needed -->
    <operation name="Tracking">
      <participant name="DocumentTracking"/>
      <output message="msg-to-service">
        <assign to="State" from="'TESTING'"/>
       <assign to="tracking-scope" from="'SYSTEM'"/>
        <assign to="." from="PrimaryDocument"/>
      </output>
      <input message="msg-from-service">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <!-- Do some work here -->
    <!-- Update state again -->
    <operation name="Tracking">
      <participant name="DocumentTracking"/>
      <output message="msg-to-service">
        <assign to="State" from="'TEST COMPLETE'"/>
       <assign to="tracking-scope" from="'SYSTEM'"/>
        <assign to="." from="PrimaryDocument"/>
      </output>
      <input message="msg-from-service">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

# Chapter 53. Document XPath Replace Service

The Document XPath Replace service enables you to replace the text of a text node in an XML document by specifying an XPath expression.

The following table provides an overview of the Document XPath Replace service:

| System name | DocXPathReplace Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, System, Internet B2B > Transora |
| Description | The Document XPath Replace service performs text replacements in the document using XPath expressions. |
| Usage example | Can be used to dynamically replace static text node identifiers with run-time configurable identifiers. |
| Preconfigured? | An instance of this service is created upon installation. There are no instance configuration variables, but you must configure the workflow variables for the service in the GPM when you are creating a business process. |
| Requires third party files? | No |
| Platform availability | All supported platforms |
| Related services | None |
| Application requirements | No |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Business process context considerations | By default, any DOCTYPE tag found in the document is removed. If you want to retain the DOCTYPE tag, see *How the Document XPath Replace Service Works*. |
| Returned status values | Returned status values:<br><br>• Success – Completed successfully.<br><br>• Error – Errors were encountered during text replacement. Check advanced status, status reports, or systems logs. |
| Restrictions | Must be a well-formed XML document that can be parsed. |

## How the Document XPath Replace Service Works

The Document XPath Replace service parses a document into a DOM (Document Object Model) so that the specified XPath expression can be evaluated and the resulting node(s) replaced with the specified value(s). By default, the Document XPath Replace service will use the current primary document unless the **documentKey** parameter is specified to point to a different document. Also by default, and to maintain backward compatibility, the Document XPath Replace service removes any DOCTYPE tag found in the document.

If you need to retain the DOCTYPE tag, there are two ways to do it.

**319**

1. The recommended way is to set the **noValidate** parameter to `true` in the BPML. Using this option simply turns off all validation except for "well formed" validation and has the additional benefit of being able to utilize document streaming.
2. The other option is to set the **keepDocType** parameter to `true` in the BPML. Using this option will still remove the DOCTYPE tag before parsing, but will re-add it before returning the updated document.

Starting with Sterling Integrator version 4.0, the Document XPath Replace service also has the ability to perform multiple (batch) replacements with just one call to the adapter instead of separate individual calls.

## Implementing the Document XPath Replace Service

To implement the Document XPath Replace service, simply add the existing instance to your business process using the GPM and configure the workflow parameters appropriately.

## Configuring the Document XPath Replace Service

To configure the Document XPath Replace service, you must specify the following field settings (workflow parameters) in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| debug | Turns on debugging for this workflow which logs extra messages to the system log. Optional. Valid values are Yes (true) and No (false). Default is No (false). |
| documentKey | Name of the document in the workflow context in which to replace text. Optional. If no document name is specified in this parameter, the primary document is used. |
| keepDocEncoding | Allows you to maintain the original document encoding for your XML documents to your generated document. For example, if you use an XML document with an encoding attribute of <?xml version= '1.0' encoding= 'UTF-16"?>, you can keep this encoding attribute value for your generated document by selecting **Yes** in the GPM for **keepDocEncoding** or by adding the following line if you are editing the BPML, <assign to="keepDocEncoding">true</assign>. After the Document XPath Replace service runs, the original header of <?xml version= '1.0' encoding= 'UTF-16"?> will be the header for the generated document. If you do not provide this parameter, the document will be encoded using "UTF-8" as the default value and a header of <?xml version= '1.0'> will be given to the generated document. Valid values are Yes (true) and No (false). Optional. |
| keepDocType | Maintains the DOCTYPE tag by stripping the tag before performing any XPath replacements and then re-adding it back to the document before returning. Large file support cannot be utilized with this parameter. Optional. Valid values are Yes (true) and No (false). Default is No (false). |

| Field | Description |
|---|---|
| noValidate | Disables any parser validation to keep from removing any DOCTYPE tags. This field overrides any use of the 'keepDocType' parameter. Using this field has the additional benefit of using document streaming. Optional. Valid values are Yes (true) and No (false). Default is No (false). |
| prefix | Prefix used with a unique identifier as the replacement text. If performing more than one replacement (batch), this field must be sequentially numbered to match up with the corresponding textNodeXPath field. Optional, but either prefix or replacmentText must be specified. |
| replacementText | Text to replace in the document. If performing more than one replacement (batch), this field must be sequentially numbered to match up with the corresponding textNodeXPath field. Optional, but either prefix or replacmentText must be specified. |
| replaceMultiple | Whether multiple occurrences of the XPath statement should be replaced or just the first it finds. If performing more than one replacement (batch), this field must be sequentially numbered to match up with the corresponding textNodeXPath field. Optional. Valid values are Yes (true) and No (false). Default is No (false). |
| textNodeXPath | XPath identifying the text node whose value needs to be replaced. Required. |

**Business Process Examples**

This example performs a single text replacement.

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"/>
    <assign to="textNodeXPath">//some/tag/text()</assign>
    <assign to="replacementText" from="'new text'"/>
  </output>
  <input message="inmsg">
    <assign to="." from="*"/>
  </input>
</operation>
```

This example performs a single text replacement, but does it for multiple occurrences:

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath" from="//some/tag/text()" />
    <assign to="replacementText" from="'new text'"/>
    <assign to="replaceMultiple">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs a single text replacement, but does it for multiple occurrences and generates a unique identifier with the supplied prefix of CMD- as the replacement text.

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath" from="//some/tag/text()" />
    <assign to="prefix">CMD-</assign>
    <assign to="replaceMultiple">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs a single text replacement and one multiple occurrence text replacement:

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath1" from="//some/tag/text()" />
    <assign to="replacementText1" from="'new text'"/>
    <assign to="textNodeXPath2" from="//some/trans/tag/text()" />
    <assign to="prefix2">TRANS-</assign>
    <assign to="replaceMultiple2">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs two different text replacements, both with multiple occurrences and both generating unique identifiers with the corresponding prefix as the replacement text:

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"></assign>
    <assign to="textNodeXPath1" from="//some/cmd/tag/text()" />
    <assign to="prefix1">CMD-</assign>
    <assign to="replaceMultiple1">true</assign>
    <assign to="textNodeXPath2" from="//some/trans/tag/text()" />
    <assign to="prefix2">TRANS-</assign>
    <assign to="replaceMultiple2">true</assign>
  </output>
  <input message="inmsg">
    <assign to="." from="*"></assign>
  </input>
</operation>
```

This example performs eight different text replacements and uses the noValidate parameter, which keeps the parser from validating any schema or DTD and does not remove the DOCTYPE tag (if any):

```
<operation name="ReplaceText">
  <participant name="DocXPathReplace"/>
  <output message="outmsg">
    <assign to="." from="*"/>
    <assign to="noValidate">true</assign>
    <!-- Update the primary document with current year -->
    <assign to="textNodeXPath1">//CNTROLAREA/DATETIME/YEAR/text()</assign>
    <assign to="replacementText1" from="substring(formattedTime, '1', '4')"/>
```

```
    <!-- Update the primary document with current month -->
    <assign to="textNodeXPath2">//CNTROLAREA/DATETIME/MONTH/text()</assign>
    <assign to="replacementText2" from="substring(formattedTime, '5', '2')"/>
    <!-- Update the primary document with current day -->
    <assign to="textNodeXPath3">//CNTROLAREA/DATETIME/DAY/text()</assign>
    <assign to="replacementText3" from="substring(formattedTime, '7', '2')"/>
    <!-- Update the primary document with current hour -->
    <assign to="textNodeXPath4">//CNTROLAREA/DATETIME/HOUR/text()</assign>
    <assign to="replacementText4" from="substring(formattedTime, '9', '2')"/>
    <!-- Update the primary document with current minute -->
    <assign to="textNodeXPath5">//CNTROLAREA/DATETIME/MINUTE/text()</assign>
    <assign to="replacementText5" from="substring(
formattedTime, '11', '2')"/>
    <!-- Update the primary document with current second -->
    <assign to="textNodeXPath6">//CNTROLAREA/DATETIME/SECOND/text()</assign>
    <assign to="replacementText6" from="substring(
formattedTime, '13', '2')"/>
    <!-- Update the primary document with current subsecond -->
    <assign to="textNodeXPath7">//CNTROLAREA/DATETIME/SUBSECOND/text()
</assign>
    <assign to="replacementText7" from="substring(
formattedTime, '15', '4')"/>
    <!-- Update the primary document with current timezone -->
    <assign to="textNodeXPath8">//CNTROLAREA/DATETIME/TIMEZONE/text()
</assign>
    <assign to="replacementText8" from="timezoneOffsetFromGMT/text()"/>
  </output>
  <input message="inmsg">
    <assign to="." from="*"/>
  </input>
</operation>
```

# Chapter 54. Dynamic Services

Dynamic services enable an application to consume Web services as it does other services—as part of a business process. Services are created based on the WSDL that you check in to the application. These services can then communicate with a specified Web service using SOAP messages as part of a business process, enabling you to extend your business to use external Web functionality.

The following table provides an overview of Dynamic services.

| System Name | Determined by the information in the WSDL. |
|---|---|
| Graphical Process Modeler (GPM) categories) | Dynamic Services |
| Description | Enables the application to consume a Web service as a normal service. User can check in the WSDL of the consumed Web service into the application. Once checked in, the application creates a service definition and service instances corresponding to operations within the WSDL and adds those service instances in the GPM Palette. The user can then use these services as a normal service and build a BPML out of it. Once this BPML is executed these services are called, which in turn makes a SOAP call to the service endpoint and returns the response back to the BPML. The main objective of the Dynamic service is to generate the SOAP message from the checked-in WSDL file. While the Default Transport configuration determines the transport to use to send the SOAP message (HTTP), a newer configuration option ("Other Transport") is transport-independent. |
| Business usage | Include Web service functionality in a business process. |
| Usage example | You need to do currency conversions in a business process, and are aware of a Web service that performs conversions on the fly. You can include a Dynamic service in a business process. The service sends an amount and currency code to the Web service, along with the code of the currency to convert to, and receives the converted value back from the Web service. |
| Preconfigured? | No. A Dynamic service is created by checking in a WSDL file. |
| Requires third party files? | WSDL, and if required for signing SOAP messages, certificates and security tokens |
| Platform availability | All supported platforms for the application |
| Related services | Related services:<br><br>• SOA Inbound Message service<br><br>• SOA Outbound Message service<br><br>• Services and adapters used for transport, such as the HTTP Client Begin Session service, HTTP Client POST service, and the HTTP Client End Session service Note: These are configured automatically for the default (legacy) HTTP transport option<br><br>• SOA Inbound Security service (optional)<br><br>• SOA Outbound Security service (optional) |

| Application requirements | Application Web services |
|---|---|
| Initiates business processes? | No. Dynamic services are used within a business process. |
| Invocation | As part of a business process |
| Business process context considerations | Requires that values for parameters are passed through workflow context to the service; once the Web service returns a result, the result becomes the primary document. |
| Returned status values | Returned status values:<br>• Success<br>• Error |
| Restrictions | None |
| Persistence level | System default |
| Testing considerations | To test a Dynamic service, include it in a business process and run the process. For an overview of the procedure, see *Implementing a Dynamic Service*. |

## How Dynamic Services Are Created

The application creates Dynamic services based on information provided in a WSDL file that you check in.

In the WSDL, you specify each Web service and port combination that is required. The application creates a GPM stencil category for each unique Web service and port combination.

The name of the stencil category will always be in upper case to avoid any conflicts with operating system restrictions. Therefore, you cannot specify two categories with the same name but with different capitalization.

In the WSDL, you also specify any operations needed for each Web service and port. The application creates a service definition and configuration of the service definition for each operation on the Web service and port.

Once created, the services are displayed in the GPM and can be used in business processes. If the Other option is used for HTTP or HTTPS transport, configuration of additional services (SOAInboundMessageProcessing, SOAOutboundMessageProcessing, and optionally SOAInboundSecurity and SOAOutboundSecurity) is required.

**Note:** Only SOAP bindings are considered for Dynamic services. Other types of bindings are ignored and Dynamic services are not created for them.

Once the Dynamic Service creation process is done, you can begin using the services in your business processes. The dynamically created services can only be used as part of a business process.

## What Happens During a Business Process

When a business process is invoked, the Dynamic service loads the WSDL (which has been stored in a cache by the check in process), creates a SOAP request for the specified operation, and posts it to the endpoint specified in the WSDL. The Dynamic service then waits for the response from the service endpoint. Once the endpoint returns a response, the Dynamic service loads the response as the

primary document, and into the work flow context, so that next service in the business process can take appropriate action on the returned response.

## Example

Your corporation has a Web service that performs currency conversions. You create a business process that receives values in U.S. dollars, but needs to have the amounts converted to Euros for a later step in the process.

You define WSDL for the Web service, specifying the information necessary to dynamically create a service, and check in the WSDL.

The application validates the WSDL and creates the new service and a stencil and category for it in the GPM. It adds the service definition to the list of services available in the application for creation, editing, and copying. It creates as many configurations of the service as needed for the functions you specify in the WSDL. So, if you specified "Convert" as the Web service to go to in the WSDL and "US_to_Euro" and "Euro_to_Yen" as the actions needed, a service category called Convert would be created on the Dynamic Services stencil. Within that new category, one service definition and one service configuration would be created for each action – US_to_Euro and Euro_to_Yen.

You include the US_to_Euro service in your business process. The US_to_Euro service sends a SOAP message containing the value (the U.S. dollar amount, say $50) to the Convert Web service. The Web service does the conversion to Euros, and sends back the converted amount to the US_to_Euro service, which passes the new Euro amount to the primary document. The next step in the business process uses the Euro amount for its operation.

You must have Web services for your application to use this functionality. You create a WSDL file that provides information about the service (or services), and the application creates the services and makes them available in the GPM for you to configure.

## Implementing a Dynamic Service

To implement a Dynamic service, complete the following tasks:
1. Obtain the WSDL for the Web service you want to access during a business process.
2. Verify that the WSDL contains the necessary content to describe the Dynamic service to the application. If it does not, you must edit the WSDL to include the required information. See *About WSDL*.
3. Check in the WSDL.
4. In the GPM, create your business process and select the desired service from the newly created stencil, then select a configuration of the service.
5. Specify the parameter settings for the service configuration.
6. Check in, test, and use the business process.

## Configuring a Dynamic Service

You must specify field settings in the application, using the GPM.

## Setting Up the Service in the GPM

The four security-related parameters in the table (Certificate, InsertSecurityHeader, SecurityToken, X509CertificateOption) are always displayed in the GPM for all Dynamic services.

Additionally, a Dynamic service can have parameters based on the partType element in the WSDL. If the partType element is from the XML Schema namespace (simple type), then these parameters will also be displayed in GPM. This normally happens for the RPC/Literal style WSDL. However, if the partType of these parameters refers to another namespace, then these parameters are not displayed in the GPM service editor. This normally happens for the Document/Literal style WSDL.

| Field | Description |
|---|---|
| Config | Select the name of the service configuration from the list. |
| Certificate | Enter the name of the digital certificate to use for signing the SOAP request. This certificate must have already been imported to the Web service endpoint system. |
| InsertSecurityHeader | Whether to insert the security header. Valid values are true and false. |
| SecurityToken | Name of the security token. This token must have already been imported to the Web service endpoint system. |
| X509CertificateOption | Used when inserting a digital certificate into the SOAP request. Valid values are:<br>• Binary Token<br>• IssuerSerial (default)<br>• X509KeyIdentifier |

## Business Process Example – RPC/Literal

The following example illustrates how a Dynamic service could be used in a business process:

You need to access a Web service that does simple mathematical calculations during a business process step. The Web service is called Calculator.

The WSDL for Calculator is shown in the following example.
1. Check in the WSDL to the application.
2. Create a business process and include one or more configurations of the new service in it.
3. Check in the business process.
4. Test the Dynamic service by running the business process. If there is a problem accessing the Web service, error messages are displayed in the BP Monitor.

The following example shows a WSDL file with the name Calculator.

The following example shows creating a Dynamic service creation against a WSDL with one port type and with two operations:

```
<?xml version="1.0" encoding="UTF-8"?>
 <wsdl:definitions name="CalculatorWSDL" targetNamespace=
      "http://00.00.00.00:1234/axis/Calculator.jws"
```

```
xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="http://00.00.00.00:1234/axis/Calculator.jws"
xmlns:intf="http://00.00.00.00:1234/axis/Calculator.jws"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:message name="subtractResponse">
   <wsdl:part name="subtractReturn" type="xsd:int"/>
  </wsdl:message>
  <wsdl:message name="addResponse">
   <wsdl:part name="addReturn" type="xsd:int"/>
  </wsdl:message>
  <wsdl:message name="addRequest">
   <wsdl:part name="i1" type="xsd:int"/>
   <wsdl:part name="i2" type="xsd:int"/>
  </wsdl:message>
  <wsdl:message name="subtractRequest">
   <wsdl:part name="i1" type="xsd:int"/>
   <wsdl:part name="i2" type="xsd:int"/>
  </wsdl:message>
  <wsdl:portType name="Calculator">
    <wsdl:operation name="add" parameterOrder="i1 i2">
      <wsdl:input message="intf:addRequest" name="addRequest"/>
 <wsdl:output message="intf:addResponse" name="addResponse"/>
   </wsdl:operation>
   <wsdl:operation name="subtract" parameterOrder="i1 i2">
     <wsdl:input message="intf:subtractRequest" name="subtractRequest"/>
     <wsdl:output message="intf:subtractResponse" name="subtractResponse"/>
   </wsdl:operation>
 </wsdl:portType>
 <wsdl:binding name="CalculatorSoapBinding" type="intf:Calculator">
    <wsdlsoap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
   <wsdl:operation name="add">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="addRequest">
     <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/"
          namespace="http://DefaultNamespace" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="addResponse">
     <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/"
          namespace="http://00.00.00.00:1234/axis/Calculator.jws"
use="encoded"/>
    </wsdl:output>
   </wsdl:operation>
  <wsdl:operation name="subtract">
    <wsdlsoap:operation soapAction="foo"/>
    <wsdl:input name="subtractRequest">
     <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/"
          namespace="http://DefaultNamespace" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="subtractResponse">
     <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/"
          namespace="http://00.00.00.00:1234/axis/Calculator.jws"
use="encoded"/>
    </wsdl:output>
   </wsdl:operation>
  </wsdl:binding>
 <wsdl:service name="CalculatorService">
    <wsdl:port binding="intf:CalculatorSoapBinding" name="IQuizService">
    <wsdlsoap:address location="http://00.00.00.00:1234/axis/
```

```
Calculator.jws"/>
     </wsdl:port>
 </wsdl:service>
 </wsdl:definitions>
```

When the file is checked in, the application attempts to validate this WSDL file. On successful validation, it creates services in the GPM stencil:

- Dynamic Services > CALCULATOR (name of the WSDL file)
- Dynamic Services > CALCULATOR > IQUIZSERVICE (Port Name)
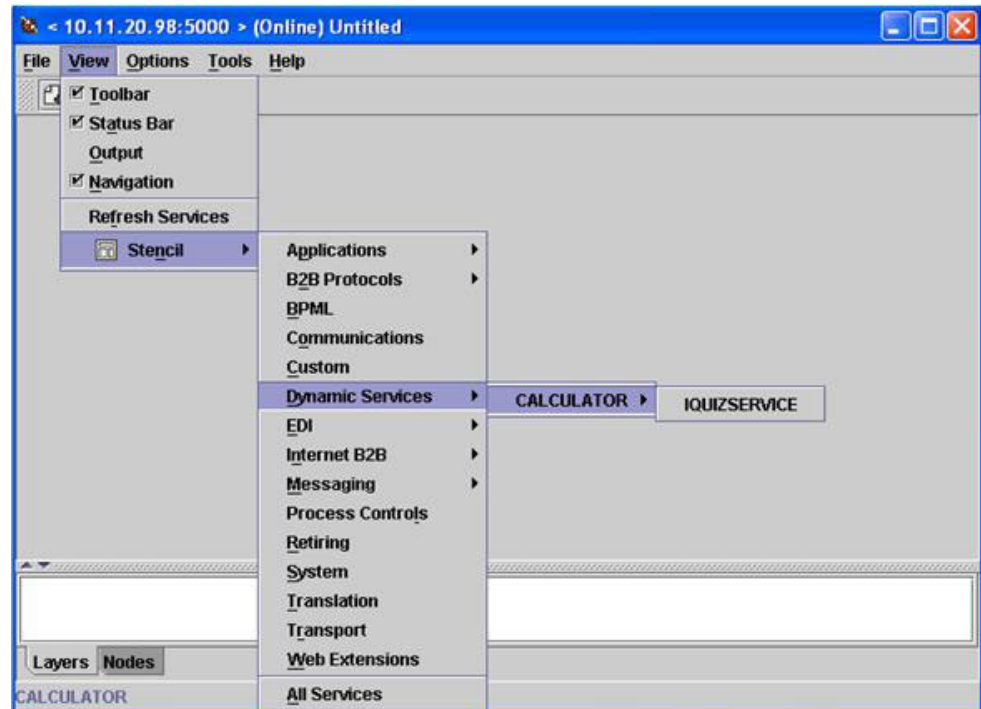


For each operation listed under a particular port, a service definition file is created in the application. For example, for the two operations (add and subtract) in the WSDL shown in the previous example, the following two service definition files are created:

- CALCULATOR_add
- CALCULATOR_subtract

A new stencil will open that will then display all the operations this port type will have as a service for the application.

The add and subtract operations are displayed as services for the IQUIZSERVICE stencil in the GPM , as shown in the following example:
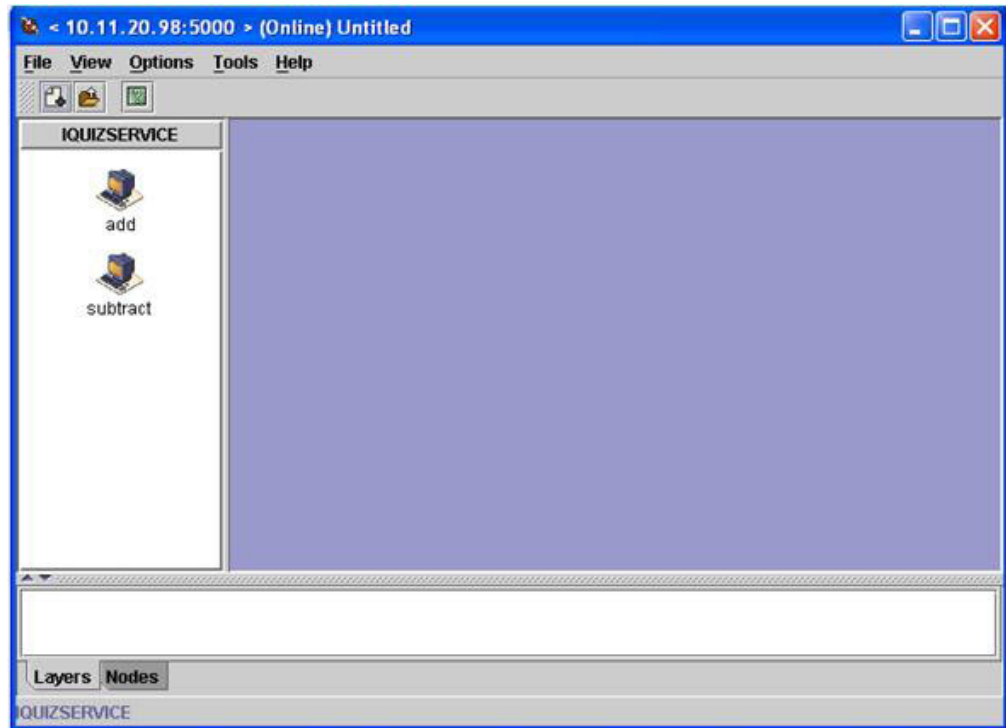
## Using a Dynamic Service in a Business Process

Once a Dynamic service is created, you can include it in a business process, as shown in the following example. Note that the add service icon is selected. In the service editor, the default service instance (DS_CALCULATOR_PORT1_OPE2_Instance) has been selected, and its parameters are displayed in the editor in the lower half of the screen. Values have been assigned for the i1 and i2 part name parameters



The Certificate, InsertSecurityHeader, SecurityToken and X509CertificateOption parameters appear in every Dynamic service configuration, but are only used when signing the SOAP request. The remaining two parameters, i1 and i2, are

specific to this service. These are the part names of the input message of the add operation of the Calculator WSDL. Values have been entered for them. (Normally, these values would be passed to the service by an earlier step in the business process.)

**Note:** The parameters for a Dynamic service are displayed in the GPM only when the partType is from the XML schema namespace (simple type). If the partType refers to another namespace, the parameters do not display in the GPM and it must be passed in to the service as a document.

Validate, save, and check in the business process to the application. When the business process runs, the add service is invoked, reads the WSDL (which has been stored in a cache by the check in process), generates a SOAP message, sends it to the Web service, and waits for a response. The Web service returns the response, it becomes the primary document (and therefore available to subsequent steps in the business process), and the business process continues.

### Request by Dynamic Service to Web Service

The following example shows a request:

```
<soapenv:Envelope xmlns:soapenc="
http://schemas.xmlsoap.org/soap/encoding/"
          xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
          xmlns:tns0="http://DefaultNamespace"
          xmlns:xs="http://www.w3.org/2001/XMLSchema"
          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body
             soapenv:encodingStyle="
http://schemas.xmlsoap.org/soap/encoding/">
       <tns0:add>
           <i1 xsi:type="xs:int">20</i1>
           <i2 xsi:type="xs:int">30</i2>
       </tns0:add>
   </soapenv:Body>
 </soapenv:Envelope>
```

### Response from the Web Service

The following is the response received by the add service from the Web service:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <soapenv:Body>
       <ns1:addResponse
           soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
           xmlns:ns1="http://DefaultNamespace">
          <addReturn xsi:type="xsd:int">50</addReturn>
       </ns1:addResponse>
    </soapenv:Body>
 </soapenv:Envelope>
```

### Errors Generated

If any input was invalid, or if there is an internal processing error at the Web service endpoint, a SOAP_FAULT will be returned. The following is an example of a SOAP fault response for invalid input obtained from the Web service endpoint:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

```
      <soapenv:Body>
    <soapenv:Fault>
         <faultcode>soapenv:Server.userException</faultcode>
         <faultstring>java.lang.NumberFormatException: z20</faultstring>
         <detail>
         <ns1:hostname
             xmlns:ns1="http://xml.apache.org/axis/">ADT</ns1:hostname>
         </detail>
         </soapenv:Fault>
      </soapenv:Body>
 </soapenv:Envelope>
```

If the endpoint specified in the WSDL is not running, then after the waiting time specified in the HTTP Client adapter, the business process at the client end will return the error NO RESPONSE FROM ENDPOINT.

## Dynamic Service for Document/Literal style WSDL

The following is an example of a Document/Literal style WSDL with one portType and one operation:

```
<?xml version="1.0" encoding="UTF-8"?>
 <wsdl:definitions name="BeadInquiryWS"
     targetNamespace="http://www.sterlingcommerce.com/mesa"
     xmlns:mesa="http://www.sterlingcommerce.com/mesa"
     xmlns:mesa_xsd="http://www.sterlingcommerce.com/mesa/schema"
     xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
     xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
     xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
     <wsdl:types>
        <xs:schema attributeFormDefault="unqualified"
            elementFormDefault="qualified"
            targetNamespace="http://www.sterlingcommerce.com/mesa/schema"
            xmlns="http://www.sterlingcommerce.com/mesa/schema"
            xmlns:tns="http://www.sterlingcommerce.com/mesa/schema"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">
        <xs:complexType name="Binary">
                <xs:simpleContent>
                    <xs:extension base="xs:base64Binary">
                         <xs:attribute name="href" type="xs:anyURI"/>
                    </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
            <xs:element name="attachment" type="tns:Binary"/>
            <xs:element name="inlineAttachment" type="xs:base64Binary"/>
        <xs:complexType name="ProcessData">
                <xs:sequence>
                    <xs:any/>
                </xs:sequence>
        </xs:complexType>
        <xs:element name="mesaFault" type="tns:MESAFault"/>
        <xs:complexType name="MESAFault">
        <xs:sequence>
                    <xs:element name="code"/>
                    <xs:element name="message"/>
                    <xs:element name="statusReport"/>
         </xs:sequence>
         </xs:complexType>
         <xs:element name="processData" type="tns:ProcessData"/>
           <xs:element name="documents">
               <xs:complexType>
                 <xs:sequence>
                     <xs:element maxOccurs="unbounded"
ref="tns:attachment"/>
                 </xs:sequence>
```

```
                    </xs:complexType>
                </xs:element>
            <xs:simpleType name="HashType">
        <xs:restriction base="xs:string">
                    <xs:enumeration value="MD5"/>
                        <xs:enumeration value="NONE"/>
                    </xs:restriction>
                </xs:simpleType>
            <xs:complexType name="MESAAuth">
                <xs:sequence>
                    <xs:element name="principal"/>
                    <xs:element name="auth">
                        <xs:complexType>
                            <xs:simpleContent>
                                <xs:extension base="xs:string">
                            <xs:attribute name="hashType"
type="tns:HashType"/>
                                </xs:extension>
                            </xs:simpleContent>
                        </xs:complexType>
                    </xs:element>
                </xs:sequence>
            </xs:complexType>
            <xs:element name="mesaAuth" type="tns:MESAAuth"/>
        <xs:element name="Bead_Inquiry" type="tns:ProcessData"
            xmlns:xs="http://www.w3.org/2001/XMLSchema"/>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="MESAResponse">
        <wsdl:documentation xmlns:wsi="
http://ws-i.org/schemas/conformanceClaim/">
            <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
        </wsdl:documentation>
        <wsdl:part element="mesa_xsd:processData" name="parameters"/>
        <wsdl:part element="mesa_xsd:attachment" name="attachment"/>
    </wsdl:message>
    <wsdl:message name="Bead_Inquiry">
        <wsdl:documentation xmlns:wsi="
http://ws-i.org/schemas/conformanceClaim/">
            <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
        </wsdl:documentation>
        <wsdl:part element="mesa_xsd:mesaAuth" name="header"/>
        <wsdl:part element="mesa_xsd:Bead_Inquiry" name="parameters"/>
    </wsdl:message>
    <wsdl:message name="MESAFault">
        <wsdl:documentation xmlns:wsi="
http://ws-i.org/schemas/conformanceClaim/">
            <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
        </wsdl:documentation>
        <wsdl:part element="mesa_xsd:mesaFault" name="parameters"/>
    </wsdl:message>
<wsdl:message name="GISGeneric">
        <wsdl:documentation xmlns:wsi="
http://ws-i.org/schemas/conformanceClaim/">
            <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
        </wsdl:documentation>
        <wsdl:part element="mesa_xsd:mesaAuth" name="header"/>
        <wsdl:part element="mesa_xsd:processData" name="parameters"/>
    </wsdl:message>
  <wsdl:portType name="GISPortType">
        <wsdl:documentation xmlns:wsi="
http://ws-i.org/schemas/conformanceClaim/">
            <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
        </wsdl:documentation>
        <wsdl:operation name="executeBead_Inquiry">
            <wsdl:documentation xmlns:wsi="
http://ws-i.org/schemas/conformanceClaim/">
```

```
                <wsi:Claim conformsTo="
http://ws-i.org/profiles/basic/1.1"/>
            </wsdl:documentation>
            <wsdl:input message="mesa:Bead_Inquiry"/>
            <wsdl:output message="mesa:MESAResponse"/>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="GISBinding" type="mesa:GISPortType">
        <wsdl:documentation xmlns:wsi="
http://ws-i.org/schemas/conformanceClaim/">
            <wsi:Claim conformsTo="http://ws-i.org/profiles/basic/1.1"/>
        </wsdl:documentation>
        <soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="executeBead_Inquiry">
            <soap:operation soapAction="sii:Bead_Inquiry"/>
            <wsdl:input>
                <soap:body parts="parameters header" use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <mime:multipartRelated>
                    <mime:part>
                      <soap:body parts="parameters" use="literal"/>
                    </mime:part>
                    <mime:part>
                      <mime:content part="attachment"
                             type="application/octetstream"/>
                    </mime:part>
                </mime:multipartRelated>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="BeadInquiryWS">
        <wsdl:port binding="mesa:GISBinding" name="GISPort">
          <soap:address location="http://10.11.20.98:5040/soap?
                 service=BeadInquiryWS"/>
</wsdl:port>
    </wsdl:service>
 </wsdl:definitions>
```

Once the file is checked in and validated, the services are created in the GPM stencil. This WSDL file, beadInquiry, would have the following entries in the GPM :

- Dynamic Services > BEADINQUIRY (name of the WSDL file)
- Dynamic Services > BEADINQUIRY > GISPORT (Port Name)

The application creates a new service definition called BEADINQUIRY_executeBead_Inquiry and a default service instance called DS_BEADINQUIRY_PORT1_OPE1_Instance .

The main difference between execution of an RPC/Literal-based Dynamic service and a Document/Literal-based Dynamic service is that the Document/Literal requires many documents to be present in the workflow context of the executed business process. For example, to execute the executeBead_Inquiry service, the workflow context of the executed business process should contain the following two documents, one each for the mesaAuth and Bead_Inquiry element:

- MesaAuth.txt:

```
<mesa:mesaAuth xmlns:mesa="http://www.sterlingcommerce.com/mesa">
<principal>admin</principal>
<auth hashType="NONE">password</auth>
</mesa:mesaAuth>
```

- BeadInquiry.txt :

```
        <mesa:Bead_Inquiry xmlns:mesa="http://www.sterlingcommerce.com/mesa"/>
```

Both elements are part of the input message of the executeBead_Inquiry operation of the checked-in WSDL file – the number and type of these required documents depends on the input message of the operation.

Before execution of any document/literal based Dynamic service, all documents mentioned in the input message of the operation should be present in the workflow context.

How these documents will be available in the workflow context depends on how you write the business process. You can write some business specific services or adapters, or can use some system level services or adapters to put these prerequisite documents into the workflow context. For example, to execute the executeBead_Inquiry service, this example uses the File System adapter to populate the two documents in the workflow context .

The first File System adapter collects the BeadInquiry.txt file from the file system and the second collects the mesaAuth.txt file from the file system. Once these two documents are in the workflow context, you can extract the two documents from the primary document and assign them to Bead_Inquiry and mesaAuth elements using the two Assign services, as shown in the following example.



The following BPML example shows how the business process might look:
```
<process name="default">
  <sequence>
    <operation name="File System Adapter">
      <participant name="AS2Extract"/>
      <output message="FileSystemInputMessage">
        <assign to="attachFile">
/server1/share/joeuser/20060307/BeadInquiry.txt
        </assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <assign to="Bead_Inquiry"
from="//PrimaryDocument/@SCIObjectID"></assign>
    <operation name="File System Adapter">
      <participant name="AS2Extract"/>
      <output message="FileSystemInputMessage">
        <assign to="attachFile">/server1/joeuser/20060307/MesaAuth.txt
</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <assign to="mesaAuth" from="//PrimaryDocument/@SCIObjectID"></assign>
<operation name="executeBead_Inquiry">
      <participant name="DS_BEADINQUIRY_PORT1_OPE1_Instance"/>
      <output message="DS_BEADINQUIRY_PORT1_OPE1InputMessage">
        <assign to="." from="*"></assign>
```

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

Because we are not generating a signed SOAP request, leave the four parameters blank.

Validate, save, check in, and execute the business process. When executed, the following happens:

- The executeBead_Inquiry service reads the WSDL file, identifying the operation and the input message that need to be executed.
- For each part of the input message, the service tries to obtain the corresponding document from the workflow context. For this example, the executeBead_Inquiry service will try to obtain the documents named mesaAuth and Bead_Inquiry. If the service fails to obtain these two documents, the service throws an exception that it cannot find the specified document in the process data. If it succeeds, the service generates the SOAP message, sends it to the endpoint, and waits for the response.

## Request Sent to Web Service

The following example shows the SOAP message generated by the executeBead_Inquiry service:

```
<soapenv:Envelope
           xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
           xmlns:xsd="http://www.w3.org/2001/XMLSchema"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
 <mesa:Bead_Inquiry
               xmlns:mesa="http://www.sterlingcommerce.com/mesa">
 </mesa:Bead_Inquiry>
 <mesa:mesaAuth xmlns:mesa="http://www.sterlingcommerce.com/mesa">
     <principal>admin</principal>
     <auth hashType="NONE">password</auth>
 </mesa:mesaAuth>
 </soapenv:Body>
 </soapenv:Envelope>
```

## Response from Web Service

The following example shows the response the executeBead_Inquiry service received from the Web service endpoint :

```
<soapenv:Envelope
      xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:wsi="http://ws-i.org/schemas/conformanceClaim/">
 <soapenv:Body>
 <mesa:processData
           xmlns:mesa="http://www.sterlingcommerce.com/mesa">
 <ProcessData>
     <service>BeadInquiryWS</service>
     <b2b-protocol>http</b2b-protocol>
     <transport-instance-id>MESAHttpServerAdapter_HttpServerAdapter_node1
 </transport-instance-id>
     <http-request-uri>/soap</http-request-uri>
     <transport-session-id>Mon Mar 13 11:33:49 IST 2006:26
```

```
</transport-session-id>
    <messageMode>1</messageMode>
    <wsConfig name="BeadInquiryWS">
        <certID></certID>
        <verificationCertID></verificationCertID>
    </wsConfig>
    <SOARequiredSignature>false</SOARequiredSignature>
    <EXPECT_SECURITY_HEADER>false</EXPECT_SECURITY_HEADER>
    <SOAP_URI>/soap</SOAP_URI>
    <SOAPEnvNSPrefix>soapenv</SOAPEnvNSPrefix>
  <SOAPEnvNSURI>
http://schemas.xmlsoap.org/soap/envelope/
 </SOAPEnvNSURI>
 <mesa xmlns="uri:sci">
  <Bead_Inquiry
        xmlns="http://www.sterlingcommerce.com/mesa"></Bead_Inquiry>
  </mesa>
    <serviceMode>0</serviceMode>
    <typeName>Bead_Inquiry</typeName>
    <PrimaryDocument SCIObjectID="blrgislin01:3399d4b3:109f23357e5:-4260">
    </PrimaryDocument>
    <ADD_SOAP_ENVELOPING>false</ADD_SOAP_ENVELOPING>
    <SOAPOutboundAttachments>
        <SOAPAttachment1
             Content-ID="cid:attachment=
                 1015796913_1142248458099@sterlingcommerce.com"
              SCIObjectID="blrgislin01:3399d4b3:109f23357e5:-4245">
        </SOAPAttachment1>
    </SOAPOutboundAttachments>
    <attachmentCID>
         cid:attachment=1015796913_1142248458099@sterlingcommerce.com
     </attachmentCID>
    <INSERT_SECURITY_HEADER>false</INSERT_SECURITY_HEADER>
 </ProcessData>
 </mesa:processData>
 <mesa:attachment
       xmlns:mesa="http://www.sterlingcommerce.com/mesa"
href="cid:attachment=1015796913_1142248458099@sterlingcommerce.com">
 </mesa:attachment>
 </soapenv:Body>
 </soapenv:Envelope>
--_=29258051243737204Sterling29258051243737204MOKO
 content-type: application/octet-stream
 content-id: <attachment=1015796913_1142248458099@sterlingcommerce.com>
 <?xml version='1.0' encoding='UTF-8'?>
   <result><row><OBJECT_ID>B2B_WF_OBJECT_ID_2</OBJECT_ID>
   <OBJECT_VERSION>1.0</OBJECT_VERSION>
   <OBJECT_NAME>HTTP_SEND_ENVELOPE_OFF</OBJECT_NAME>
   <TRANSPORT_DESC>HTTP</TRANSPORT_DESC>
   <ENVELOPE_DESC>NO- ENVELOPE
 </ENVELOPE_DESC>
 <MESSAGE_MODE>send</MESSAGE_MODE>
 <WORKFLOW_VERSION>1.0</WORKFLOW_VERSION>
 <OBJECT_CLASS>B2B_WORKFLOW</OBJECT_CLASS>
 <LAST_MODIFICATION></LAST_MODIFICATION>
 <LAST_MODIFIER></LAST_MODIFIER><OBJECT_STATE></OBJECT_STATE></row>
 --_=29258051243737204Sterling29258051243737204MOKO--
```

## About WSDL

For creation of a Dynamic service to succeed, the following elements need to be
included in the WSDL:

**Note:** See *WSDL Validation* for more information about specific rules used the
validation process.

| Element | Description |
|---|---|
| Service | At least one service with a name associated with it. |
| Ports | For each service, there should be one or more ports, each with a unique name. |
| SOAP port | At least one port must have a SOAP address location specified for it. |
| Endpoint | Each SOAP address must include a location attribute that specifies the endpoint where the Web service runs. The location must be a proper URI and the protocol must be HTTP or HTTPS. |
| Binding | Each port must refer to a binding element, using the binding attribute. For Dynamic services, only SOAP binding over HTTP or HTTPS is supported. If the WSDL contains both SOAP and non-SOAP bindings, Dynamic services are created for only the SOAP addresses – non-SOAP bindings are ignored. The binding element describes how the service is bound to a messaging protocol, particularly the SOAP messaging protocol. You can use either Remote Procedure Call (RPC) or Document style bindings. If the attribute is not present, then the system assumes the style is Document. The SOAP binding can also have an encoded use or a literal use. |
| Operations | The number of operations in the binding section and the port type section must be the same. The names must be the same. |
| Input/Output Messages in Operations | Each operation must have input and output operations associated with it. Dynamic Service Creation supports only Request-Response transmission and does not support solicit-response or one-way transmission types. |
| Parts | Each message can include zero or more parts. Each part must have a unique name. |
| Part Element | If the WSDL uses Document/Literal style binding, each part must have a part element that refers to a root element of the schema defined in the WSDL. |
| Duplicate Operations | WSDL files that use RPC binding can have two duplicate operations, but not more than that. WSDL that uses Document/Literal binding can never have any duplicate operations. |

## Checking In a New WSDL

If you check in a new version of a WSDL file, the following occurs:

- Old Dynamic services based on the old WSDL are deleted.
- New Dynamic services are created based on the new WSDL file

   **Note:** When a WSDL has multiple versions, the Dynamic service is created for the default version only.

## Deleting Dynamic Services

To delete a Dynamic service, delete the WSDL. When you delete a checked-in WSDL file, the application deletes the following items, in this sequence:

1. All service configurations associated with the WSDL
2. All service definitions associated with the WSDL
3. The XML file created for the service definition
4. GPM stencil categories and items associated with the WSDL

# Chapter 55. E-5 2000 Adapter

The E-5 2000 adapter enables business processes to interact with E-5 clients and servers. E-5 is a B2B standard developed by the Automotive Industry Action Group (AIAG). This adapter provides computer-to-computer delivery and receipt of EDI, XML, or other proprietary data agreed upon by the trading partners.

The E-5 2000 adapter has two primary purposes:

- Provide E-5 client-side services
- Provide E-5 server-side services

The E-5 2000 adapter uses predefined business processes to assist communication with other E-5 servers or clients. These business processes provide client and server functions for delivering a document, obtaining a resultant document, and acknowledging receipt of a resultant document.

The following table provides an overview of the E5 Client Configuration adapter:

| System name | None |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B |
| Description | The E5 Client Configuration adapter is a set of model business processes that allows a user to send and retrieve documents from an E-5 server. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | E5 Client Configuration service |
| Application requirements | The adapter supports E-5 2000 or version 2.0 of the E-5 standard. Previous versions of E-5 are not supported. |
| Initiates business processes? | No |
| Invocation | The API Definition, Deliver Submit, Obtain Submit and Acknowledge Submit run by starting the E5Client_StartSubmit.bpml. The Loop Back Test runs by starting the E5Client_LoopBackTest.bpml |

| Business process context considerations | Business process context considerations: |
|---|---|
| | • API Definition – Run E5ClientConfig. No documents in ProcessData. |
| | • Deliver Submit – Run E5ClientConfig. The Deliver Submit document should be placed as PrimaryDocument and attachment document as AttachmentDocument in ProcessData. |
| | • Obtain Submit – Run E5ClientConfig. The Obtain Submit document should be placed as PrimaryDocument in ProcessData. |
| | • Acknowledge Submit – Run E5ClientConfig. The Acknowledge Submit document should be placed as PrimaryDocument in ProcessData. |
| | • Loop Back Test – Run E5ClientConfig. The Deliver Submit document should be placed as PrimaryDocument and attachment document as AttachmentDocument in ProcessData. |
| Returned status values | If a message code of Serious is returned from the E-5 server, the E5 Client Configuration adapter sends the business process an error. |
| Restrictions | There can be multiple configurations of this service. |
| Testing considerations | See *Testing the E5 Client Configuration Adapter*. |

The following table provides an overview of the E5 Server Configuration adapter:

| System name | None |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B |
| Description | The E5 Server Configuration adapter is a set of model business processes that acts as a server by receiving and storing documents from an E-5 client. These documents can also be retrieved by E-5 clients. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | E5 Server Configuration service, E5 Obtain service |
| Application requirements | For information on how to deploy the E5 Server Configuration adapter for Sterling Integrator, see the User Manual. |
| Business process context considerations | No |
| Returned status values | When applicable error messages are returned to the E-5 client. |
| Restrictions | There may be multiple configurations of this service. |
| Testing considerations | See *Testing the E-5 Server Configuration Adapter*. |

## Requirements

To implement, configure, and use the E-5 2000 adapter, you should be familiar with:

• The E-5 2000 specification (E-5 version 2.0 or AIAG Guideline for Electronic Commerce Message Routing).

- XML concepts
- How process data and documents are handled in Sterling Integrator
- BPML concepts
- Trading partner setup
- Perimeter server configuration

For the E-5 2000 adapters to work correctly, verify that you have installed and configured a Perimeter server to work with the HTTP Server adapter.

## How the E-5 2000 Adapter Works

The E5 Client Configuration adapter is a set of business processes that can be used within Sterling Integrator to facilitate communication with an E-5 server. The E5 Client Configuration adapter uses Sterling Integrator trading profiles and the HTTP Send adapter in conjunction with the custom developed E5 Client Configuration service and other internal Sterling Integrator services to send documents and retrieve them from an outside E-5 server. The E-5 business processes are customizable examples of how a customer can use Sterling Integrator services to communicate with an E-5 server.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Client adapter and its related services. See** *Retiring and Removed Services and Adapters***.**

The Sterling Integrator E5 Client Configuration service enables you to specify parameters that are required to send E-5 requests. See *Configuring the E5 Client Configuration and E5 Server Configuration Services* for more detailed information.

The E5 Server Configuration adapter is a set of business processes used within Sterling Integrator to facilitate communication with an E-5 client. The E5 Server Configuration adapter uses the Sterling Integrator HTTP Server adapter in conjunction with the custom-developed E5 Server Configuration service, E5 Obtain service, and other internal Sterling Integrator services to receive and store documents as well as retrieve documents requested by E-5 clients. The E-5 business processes are customizable examples of how a customer can use Sterling Integrator services to communicate with an E-5 client.

The E5 Server Configuration service enables you to specify parameters that are required to respond to E-5 requests. See *E5 Server Adapter Configuration* for more detail.

The adapter will receive unsolicited messages in the proper format and start a BPML to process the incoming data. The adapter will also provide, through the services of the Correlation service and API, a repository of documents that can be searched by an E-5 client to obtain responses to delivered documents.

## E-5 Client and Server Interactions

The following example shows standard client/server interactions that this adapter supports:

```
E-5 Client                                      E-5 Server

                  Deliver doc for processing        ┌─────────────────┐
                  ─────────────────────────▶        │ Process Document │
                                                     └─────────────────┘
                  Resultant                                   │
                  Document                                    ▼
                  Request to obtain doc         ┌──────┐  ┌──────────────┐
                  ─────────────────────────▶    │Search│  │ E-5 Document │
                                                │      │  │  Repository  │
                  Return resultant doc          │      │  │              │
                  ◀─────────────────────────    │      │  │              │
                                                │      │  │              │
                  Acknowledge receipt of doc    │      │  │              │
                  ─────────────────────────▶    └──────┘  └──────────────┘
```
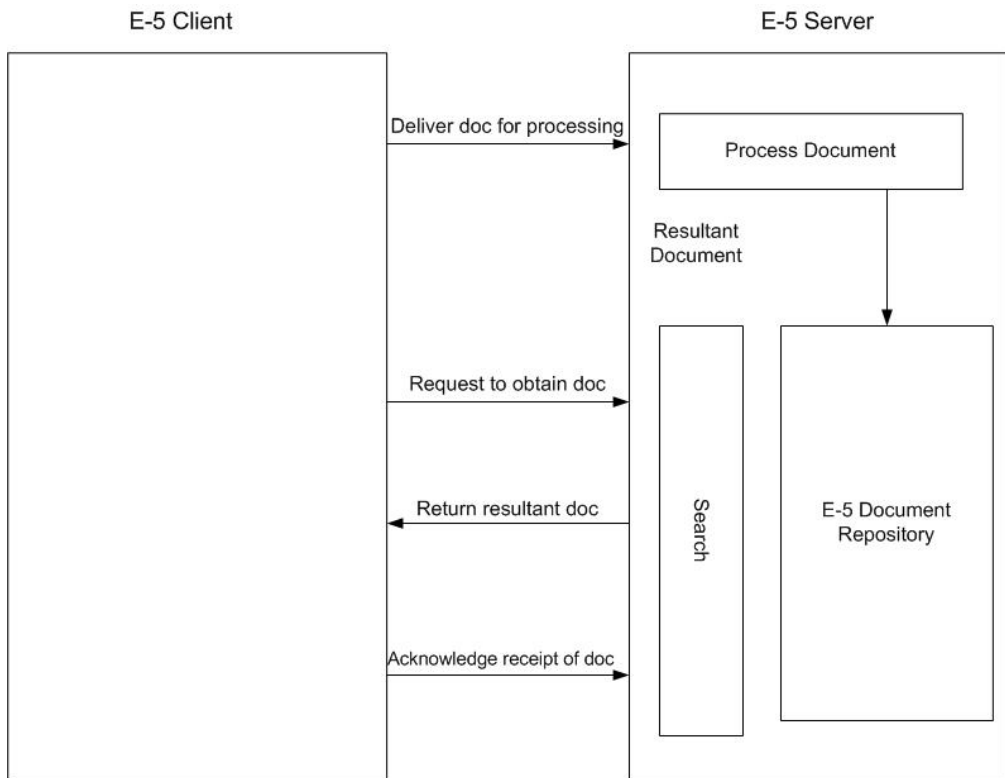
## Message Formats

E-5 communication is a synchronous request/response model. Each API command has a DTD specification for the request and the response metadata that accompanies the actual data in the transfer. The response should be received in the same HTTP session.

The following figure shows a typical message format with metadata in the form of XML and the data being transferred. All messages are packaged in MIME formatting. The E-5 2000 adapter uses the HTTP Server adapter and the HTTP Send adapter or the new HTTP Client adapter. The transport mechanism to communicate with the far-end E-5 server is HTTP or HTTP/S. Communications are secure according to those standards.
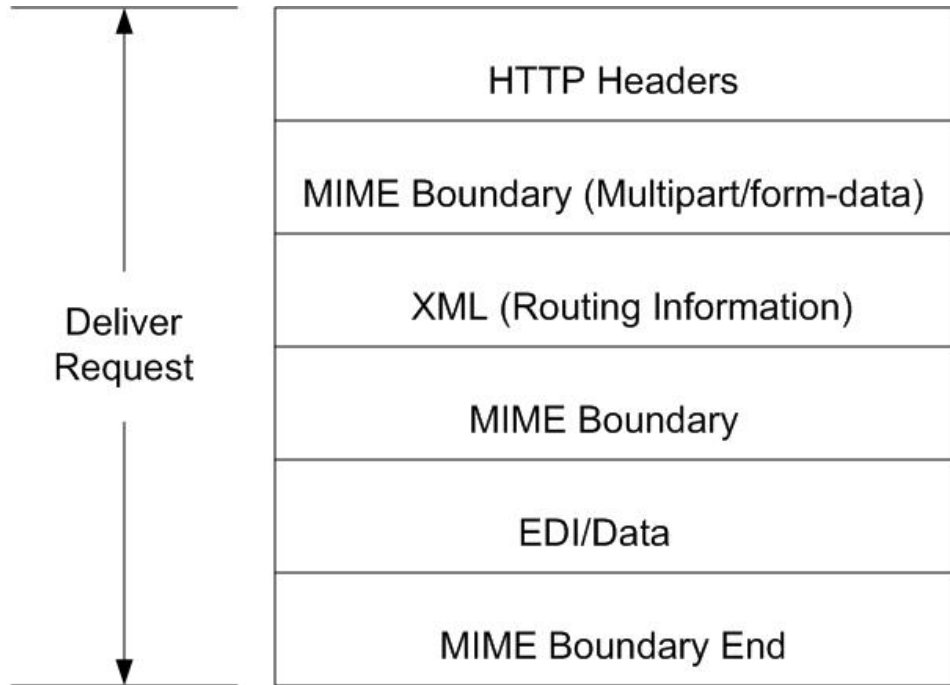
Note: Follows RFC 1867 (Form-based File Upload in HTML)

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Client adapter and its related services. See** *Retiring and Removed Services and Adapters*.

## Implementing the E-5 2000 Adapter

To implement the E-5 2000 adapter, complete the following tasks:

1. Activate your license for the E5 Client Configuration and E5 Server Configuration adapters. See *An Overview of Implementing Services*.
2. Configure trading profiles for E-5. The following sample trading partner configurations are located in the *install_dir*\e5 samples\TradPartnerProfiles folder:
   * E5TPP.xml – non-SSL trading partner configuration
   * E5SSLTPP.xml – SSL trading partner configuration
     Special configuration of the server transport is required if using SSL :
   * Specify port 443 in URL. You might need to leave the URL as HTTP instead of HTTPS, to avoid potential issues sometimes experienced when using JAVA to decode HTTPS as a URL.
   * On the Security page, select Must for SSL, STRONG for Cipher strength, and add the certificate you use
3. Configure the DTDs using the Sterling Integrator Map Editor:
   a. Load the DTDs from the *si_install*\e5 samples\dtds folder.
   b. Map the DTD corresponding to the desired request to your information.
      Special instructions for mapping the Obtain Submit DTD:

- Go to withs->between->Attributes->name. Right click on name and select properties. Select the Type tab and in the default selection click the button for Implied, the default value should now be grayed out.
- Go to withs->equals->Attributes->name. Right click on name and select properties. Select the Type tab. Select CDATA for the Attribute Type. In the default selection click the button for Implied, the default value should now be grayed out.
- Go to bys->by->Attributes->name. Right click on name and select properties. Select the Type tab. Select CDATA for the Attribute Type. In the default selection click the button for Implied, the default value should now be grayed out.

  The E-5 2000 adapter is dependent on the DTDs that are packaged with the adapter. Changes to the DTDs loaded into Sterling Integrator may cause adapter failure.
4. Create configurations for the E5 Client Configuration and E5 Server Configuration services. See *Creating a Service Configuration*.
5. Configure the E5 Client Configuration and E5 Server Configuration services. See *Configuring the E5 Client Configuration and E5 Server Configuration Services*.
6. Edit the following preconfigured configurations:
   - E5 HTTP Send adapter

     **Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Client adapter and its related services. See *Retiring and Removed Services and Adapters*.
   - E5 Lightweight JDBC adapter: Define the following configuration fields:

     Start a new business process - select This Lightweight JDBC adapter will not start a new business process

     Pool Name - select the pool name
7. Use the E-5 adapter in a business process.

## Configuring the E5 Client Configuration and E5 Server Configuration Services

To configure the E5 Client Configuration and E5 Server Configuration services, you must specify field settings in Sterling Integrator.

## E5 Client Adapter Configuration

To configure the E5 Client adapter:
1. In Sterling Integrator, edit the following fields, as appropriate:

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

   **Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the HTTP Send adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Client adapter and its related services. See *Retiring and Removed Services and Adapters*.

| Field | Description |
|---|---|
| Name | Unique and meaningful name for the service configuration. Required. |
| Description | Meaningful description for the service configuration, for reference purposes. Required. |
| Select a Group | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br><br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br><br>• 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.<br><br>**Note:** See *Using Service Groups*. |
| E5 Sending Contract Name (E5_CONTRACTID) | Choose the trading profile contract name for sending to the remote E5 server. Required. |
| E5 Remote Username (E5_USERNAME) | Remote E5 server username. Required. This username was created as part of the E5 HTTP Send adapter configuration. |
| E5 Authentication (E5_AUTHENTICATION) | To use E5 server authentication, type TRUE; otherwise, type FALSE. Required. |
| Certificate Name(s) (E5_CERTIFICATENAME) | If SSL is being used, enter the certificate names. Required if using SSL. |

2. Exchange certificates with the E-5 server if using SSL. Check in the E-5 certificates as a trusted certificate. If Sterling Integrator uses a CA certificate, check in the E-5 certificate as a CA certificate.

## E5 Server Adapter Configuration

To configure the E5 Server Configuration adapter:

1. In Sterling Integrator, edit the following fields, as appropriate:

   **Note:** It is not recommended that you create more than one instance of this service.

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

| Field | Description |
|---|---|
| Select a Group | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** See *Using Service Groups*. |
| E5 DTD Directory | Path to where the E5 DTDs are stored. Required. |
| Local E5 API URI | URI where an outside E5 server may send an E5 API Request to Sterling Integrator. Required. |
| Local URI for the E5 API Result DTD | URI where an outside E5 server may access the E5 API Result DTD on Sterling Integrator. Required. |
| Local E5 Deliver URI | URI where an outside E5 server may send an E5 Deliver Request to Sterling Integrator. Required. |
| Local URI for the E5 Deliver Submit DTD | URI where an outside E5 server may access the E5 Deliver Submit DTD on Sterling Integrator. Required. |
| Local URI for the E5 Deliver Result DTD | URI where an outside E5 server may access the E5 Deliver Result DTD on Sterling Integrator. Required. |
| Local E5 Obtain URI | URI where an outside E5 server may send an E5 Obtain Request to Sterling Integrator. Required. |
| Local URI for the E5 Obtain Submit DTD | URI where an outside E5 server may access the E5 Obtain Submit DTD on Sterling Integrator. Required. |
| Local URI for the E5 Obtain Result DTD | URI where an outside E5 server may access the E5 Obtain Result DTD on Sterling Integrator. Required. |
| Local E5 Acknowledge URI | URI where an outside E5 server may send an E5 Acknowledge Request to Sterling Integrator. Required. |
| Local URI for the E5 Acknowledge Submit URI DTD | URI where an outside E5 server may access the E5 Acknowledge Submit DTD on Sterling Integrator. Required. |
| Local URI for the E5 Acknowledge Result URI DTD | URI where an outside E5 server may access the E5 Acknowledge Result DTD on Sterling Integrator. Required. |
| Local E5 Deliver Loop Back Test URI | URI where an outside E5 server may send an E5 Deliver Loop Back Test Request to Sterling Integrator. Required. |
| E5 Template Path | Path to the E5 server templates. Required. |

2. Install and configure a Perimeter server for use with the E-5 adapter.
3. Create a configuration of the HTTP Server adapter for use with the E-5 adapter. See *Creating a Service Configuration* and *HTTP Server Adapter*, using the following E-5 specific settings for the fields indicated:
   - User Authentication field: Yes

     You must create the user on the server side, then give the username and password to the E-5 client.

- URIs field: The following table shows a sample configuration of the URIs for the HTTP Server adapter to support the E-5 server business processes. Set up your HTTP Server adapter in a similar fashion before communicating with the far end E-5 client.

| URI | BPML To Run | Raw Messages |
| --- | --- | --- |
| /b2bhttp/inbound/E5Server | E5Server_Receive_All | Yes |
| /b2bhttp/inbound/E5ServerAPI | E5Server_Receive_APIRequest | No |
| /b2bhttp/inbound/ E5ServerLoopBackTest | E5Server_LoopBackTest | Yes |
| /b2bhttp/E5_V20_APIs_Result.dtd | E5Server_DTD_Response | Yes |
| /b2bhttp/ E5_V20_Deliver_Submit.dtd | E5Server_DTD_Response | Yes |
| /b2bhttp/E5_V20_Deliver_Result.dtd | E5Server_DTD_Response | Yes |
| /b2bhttp/E5_V20_Obtain_Submit.dtd | E5Server_DTD_Response | Yes |
| /b2bhttp/E5_V20_Obtain_Result.dtd | E5Server_DTD_Response | Yes |
| /b2bhttp/ E5_V20_Acknowledge_Submit.dtd | E5Server_DTD_Response | Yes |
| /b2bhttp/ E5_V20_Acknowledge_Result.dtd | E5Server_DTD_Response | Yes |
| /b2bhttp/inbound/E5Server_Obtain | E5Server_Receive_Obtain | Yes |

## E5 Client Configuration Adapter Business Processes

The following table describes the predefined business processes associated with the E5 Client Configuration adapter:

| BPML Name | Usage Description |
| --- | --- |
| E5Client_Acknowledge_Submit.bpml | • Run by E5Client_StartSubmit.<br>• Runs E5_MIME_Encode.<br>• Tells the E-5 server to mark the document corresponding to the Transaction ID as acknowledged. |
| E5Client_Deliver_Submit.bpml | • Run by E5Client_StartSubmit.<br>• Runs E5_MIME_Encode.<br>• Sends meta data and a Document to the E-5 server. |
| E5Client_GetAPI.bpml | • Run by E5Client_StartSubmit.<br>• Runs E5Client_Release_B2B.<br>• Requests the API Definition from the E-5 server. |
| E5Client_LoopBackTest.bpml | • Run from business user when performing the E5 Loop Back Test.<br>• Runs E5_MIME_Encode.<br>• Sends meta data and a Document to the E-5 server as a communication/connection test. |
| E5Client_Obtain_Submit.bpml | • Run by E5Client_StartSubmit.<br>• Runs E5_MIME_Encode, E5_MIME_Decode_Obtain.<br>• Requests Document(s) or a list of Document(s) from the E-5 server. |

| BPML Name | Usage Description |
|---|---|
| E5Client_Release_B2B.bpml | • Run by E5Client_GetAPI.<br>• Releases the B2B parameters in ProcessData. |
| E5Client_StartSubmit.bpml | • Run from business user when performing the E5 API Definition, Deliver Submit, Obtain Submit, and Acknowledge Submit functions.<br>• Runs E5Client_GetAPI, E5Client_Deliver_Submit, E5Client_Obtain_Submit, E5Client_Acknowledge_Submit. |
| E5_MIME_Decode_Obtain.bpml | • Run by E5Client_Obtain_Submit, E5Client_LoopBackTest.<br>• Decodes multiple MIME documents. |
| E5_MIME_Encode.bpml | • Run by E5Client_Deliver_Submit, E5Client_Obtain_Submit, E5Client_Acknowledge_Submit, E5Client_LoopBackTest.<br>• Encodes MIME documents. |

## Facade BPML

Use a facade BPML to run the predefined business processes listed in *E5 Client Configuration Adapter Business Processes*. The facade contains the environmental setup information that enables the business process to communicate successfully with E-5 servers. You should customize the facade to meet the specific needs of your company.

| BPML | Description |
|---|---|
| API Definition | • Run E5ClientConfig.<br>• Place no documents in ProcessData.<br>• Run E5Client_StartSubmit.bpml. |
| Deliver Submit | • Run E5ClientConfig.<br>• Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData.<br>• Run E5Client_StartSubmit.bpml. |
| Obtain Submit | • Run E5ClientConfig.<br>• Place the Obtain Submit document as PrimaryDocument in ProcessData.<br>• Run E5Client_StartSubmit.bpml. |
| Acknowledge Submit | • Run E5ClientConfig.<br>• The Acknowledge Submit document should be placed as PrimaryDocument in ProcessData.<br>• Run E5Client_StartSubmit.bpml. |
| Loop Back Test | • Run E5ClientConfig.<br>• Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData.<br>• Run E5Client_LoopBackTest.bpml. |

# E5 Server Configuration Adapter Business Processes

The following table describes the predefined business processes associated with the E5 Server Configuration adapter:

**Note:** Business processes (instances) using this model should remain in the system for the specified amount of time to 30 days for some of the business processes, so that the documents will stay in the repository for 30 days.

| BPML Name | Usage Description |
|---|---|
| E5Server_Acknowledge.bpml | • Run by E5Server_Receive_All.<br>• Runs E5Server_Input_ErrorMsg.<br>• Records that a document has been acknowledged.<br>• Recommended to change system time to 30 days. |
| E5Server_DTD_Response.bpml | • Run by HTTP Server adapter.<br>• Provides DTDs to clients through a URI. |
| E5Server_Deliver.bpml | • Run by E5Server_Receive_All.<br>• Invokes E5Server_Input_ErrorMsg, E5Server_Deliver_ProcessDocument, E5Server_StoreValues.<br>• Stores meta data and Documents in the Correlation service. |
| E5Server_Deliver_ProcessAttachment.bpml | • Run by E5Server_Deliver_ProcessAttachment.<br>• Obtains the attachment and sets the obtain date, acknowledge date, and acknowledge flag. Business user can then process the attachment document<br>• Recommended to change system time to 30 days. |
| E5Server_Deliver_ProcessDocument.bpml | • Run by E5Server_Deliver.<br>• Invokes E5Server_Deliver_ProcessAttachment.bpml<br>• If the Business User wants to process the attachment once it is delivered, the Business User will have to modify the rule in this BP. |
| E5Server_Input_ErrorMsg.bpml | • Run by E5Server_Acknowledge, E5Server_Deliver, E5Server_Obtain, E5Server_Receive_APIRequest.<br>• Inputs Error Messages into the results templates. |
| E5Server_LoopBackTest.bpml | • Run by HTTP Server adapter.<br>• Invokes E5_MIME_Decode, E5_MIME_Encode_Obtain, E5Server_Input_ErrorMsg.<br>• Returns the documents received to the E5 client requester. |
| E5Server_Obtain.bpml | • Run by HTTP Server adapter.<br>• Invokes E5Server_Input_ErrorMsg, E5_MIME_Encode_Obtain.<br>• Retrieves documents by request.<br>• Recommended to change system time to 30 days. |
| E5Server_Receive_APIRequest.bpml | • Run by HTTP Server adapter.<br>• Invokes E5Server_Input_ErrorMsg.<br>• Returns an API Definition document. |

| BPML Name | Usage Description |
|---|---|
| E5Server_Receive_All.bpml | • Run by HTTP Server adapter.<br>• Invokes E5_MIME_Decode, E5Server_Acknowledge, E5Server_Deliver, E5Server_Input_ErrorMsg.<br>• Receives E5 client requests and sorts them according to type. |
| E5Server_StoreValues.bpml | • Run by E5Server_Deliver.<br>• Stores documents with name/value pairs in the Correlation service.<br>• Recommended to change system time to 30 days. |
| E5_MIME_Decode.bpml | • Run by E5Server_Receive_All, E5Server_Receive_Obtain, E5Server_LoopBackTest.<br>• Decodes MIME documents. |
| E5_MIME_Encode_Obtain.bpml | • Run by E5Server_Obtain, E5Server_LoopBackTest.<br>• Encodes multiple MIME documents. |
| E5Server_Receive_Obtain.bpml | • Run by HTTP Server adapter.<br>• Invokes E5Server_Obtain. |

## BPML Error Handling

There are two categories of adapter errors:
• Service failures – Can be detected by on fault BPML constructs.
• Detectable errors – Must be checked for specifically.

## Functionality Supported for Sterling Integrator

The following list contains the subset of Obtain features supported by this adapter:
• Support up to four "equals" search parameters
• Wildcard character searches are supported
• No support for "between" searching
• No support for "by" searching

Acknowledgement of Document(s) removes it from the list of possible documents to be returned in an Obtain search. This adapter follows the standard for not Obtaining Documents that have an Acknowledge DateTime set, due to contradictions in the E-5 specification.

## Testing the E5 Client Configuration Adapter

To test the E5 Client Configuration adapter:
1. Configure the E5ClientConfig service instance.
2. Using the following table, perform the steps for each E-5 operation, and then verify the results:

| E-5 Operation | Steps | Result |
|---|---|---|
| API Definition | 1. Invoke E5ClientConfig.<br>2. Place no documents in ProcessData.<br>3. Invoke E5Client_StartSubmit.bpml. | API Definition Results document should be the PrimaryDocument. |
| Deliver Submit | 1. Invoke E5ClientConfig.<br>2. Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData.<br>3. Invoke E5Client_StartSubmit.bpml. | Deliver Results document should be the PrimaryDocument. |
| Obtain Submit | 1. Invoke E5ClientConfig.<br>2. Place the Obtain Submit document as PrimaryDocument in ProcessData.<br>3. Invoke E5Client_StartSubmit.bpml. | Obtain Results documents and attachment documents, named RoutingDocDecoded_1, AttachmentDocDecoded_1, RoutingDocDecoded_2, etc. |
| Acknowledge Submit | 1. Invoke E5ClientConfig.<br>2. Place the Acknowledge Submit document as PrimaryDocument in ProcessData.<br>3. Invoke E5Client_StartSubmit.bpml. | Acknowledge results document should be the PrimaryDocument. |
| Loop Back Test | 1. Invoke E5ClientConfig.<br>2. Place the Deliver Submit document as PrimaryDocument and attachment document as AttachmentDocument in ProcessData.<br>3. Invoke E5Client_LoopBackTest.bpml. | RoutingDocDecoded_1 and AttachmentDocDecoded_1 |

## Testing the E-5 Server Configuration Adapter

To test the E-5 Server Configuration adapter:

1. Configure service instances for the E-5 Server Configuration service, E-5 Obtain service, and HTTP Server adapter.
2. Using the following table as a guide, test each of the E-5 operations by verifying that the E-5 client submitted the correct document:

| E-5 Operation | Description |
|---|---|
| API Definition | 1. The client sends the request (configured per the specifications in the AIAG Guideline for Electronic Commerce Message Routing) to the URI that has been set up to handle API requests. This request is received through the HTTP Server adapter<br><br>2. Invokes the E5Server_Receive_APIRequest.bpml.<br><br>3. An API definition document is created using a template placed in a predefined location and the document is returned to the E-5 client. |
| Deliver Submit | 1. The client sends the meta data and document according to the specifications in the AIAG Guideline for Electronic Commerce Message Routing. The request is sent to the URI that has been set up to handle general E-5 server requests and is received through the HTTP Server adapter<br><br>2. Invokes the E5Server_Receive_All.bpml.<br><br>3. The document is stored in the Correlation service along with name/value pairs based on the meta data and other parameters.<br><br>4. A Deliver results document is created using a template placed in a pre-defined location and the document is returned to the E-5 client. |
| Obtain Submit | 1. The client sends the request according to the specifications in the AIAG Guideline for Electronic Commerce Message Routing.<br><br>   Note: Bys and Betweens are not supported in this E5 server.<br><br>2. The request is sent to the URI that has been set up to handle general E-5 server requests and is received through the HTTP Server adapter.<br><br>3. Invokes the E5Server_Receive_All.bpml. The E5 Obtain service is used to search the Correlation service for the requested information.<br><br>4. The Obtain results are returned to the E-5 client. |
| Acknowledge Submit | 1. The client sends the request according to the specifications in the AIAG Guideline for Electronic Commerce Message Routing. The request is sent to the URI that has been set up to handle general E-5 server requests and is received through the HTTP Server adapter.<br><br>2. Invokes the E5Server_Receive_All.bpml. The acknowledgeFlag and acknowledgeDateTime are updated in the Correlation service.<br><br>3. The acknowledge results are returned to the E-5 client. |

# Chapter 56. ebXML BPSS Correlation Service

The following table provides an overview of the ebXML BPSS Correlation service:

| System name | BPSSCorrelation |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Interfaces with the Correlation API to track transactions, activities and documents going out and coming in as part of an ebXML message. |
| Business usage | Tracks the:<br>• BPSS transaction<br>• BPSS activities within a transaction<br>• Documents within a BPSS activity |
| Usage example | A business process that needs to track the BPSS activities can invoke this service by passing the required parameters. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Restrictions | None |

## Implementing the ebXML BPSS Correlation Service

To implement the ebXML BPSS Correlation service, complete the following tasks:

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

## Configuring the ebXML BPSS Correlation Service

To configure the ebXML BPSS Correlation service, you must define following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| conversation_id | The conversation_id of the ebXML message. Required. |
| cpa_id | A valid Collaboration Protocol Agreement (CPA) ID. Required. |

| Field | Description |
|---|---|
| action | Indicates to the service what action needs to be performed. Required. Valid values are:<br>• SELECT<br>• INSERT<br>• UPDATE |
| DBValues | The amount of information passed within the DBValues node varies depending on the action. Required. |

## ebXML BPSS Correlation Service Business Process Usage

When the action is SELECT, an attempt is made to retrieve the transaction ID. The status values shown in the following examples are:

• 0 = WAITING
• 1 = COMPLETED
• 9 = FAILED

The InDoc should contain the following values when the action is SELECT:

```
<inDoc>
   <DBValues>
      <tp_name>Sterling 2</tp_name>
      <status>0</status>
   </DBValues>
</inDoc>
```

When the action is INSERT, an attempt is made to insert the values shown in the following example:

```
<inDoc>
   <DBValues>
      <tp_name>Sterling 2</tp_name>
      <status>0</status>
      <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
      <time_to_perform>1440</time_to_perform>
      <role_name>buyer</role_name>
      <start_time>1062126842123</start_time>
      <transaction_type>IssuePurchaseOrder</transaction_type>
   </DBValues>
</inDoc>
```

When the action is UPDATE, an attempt is made to insert the values shown in the following example. The status values shown are:

• 1 = COMPLETED
• 9 = FAILED

```
<inDoc>
   <DBValues>
      <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
      <activity>RequestPurchaseOrder</activity>
      <doc_id>sgcentennial:4bdb20:f74ba603b7:-6166</doc_id>
      <ctr>1</ctr>
      <workflow_id>7015</workflow_id>
      <status>1</status>
      <end_time>1062140580230</end_time>
      <exception_workflow_id>0</exception_workflow_id>
   </DBValues>
 </inDoc>
```

## Output from Service to Business Process

The following parameters are passed from the service to a business process:

| Field | Description |
|---|---|
| OutValues | Contains the retrieved values. Only applicable when the input action parameter is SELECT. Optional. |

The following output values will only be returned if a successful selection is made:

```
<OutValues>
   <OutDoc>
      <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
      <time_to_perform>2880</time_to_perform>
   </OutDoc>
</OutValues>
```

The following parameters need to be passed to the service when it is being Run with the output message set to processActivity:

| Field | Description |
|---|---|
| action | Indicates to the service what action needs to be performed. Required. Valid values are:<br>• SELECT<br>• INSERT<br>• UPDATE |

When the action is SELECT, an attempt is made to retrieve the count of the activity name corresponding to the transaction ID:

```
<inDoc>
   <DBValues>
      <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
      <activity>RequestPurchaseOrder</activity>
   </DBValues>
</inDoc>
```

When the action is SELECT_ID, an attempt is made to retrieve the business process ID of the activity:

```
<inDoc>
   <DBValues>
      <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
      <activity>RequestPurchaseOrder</activity>
   </DBValues>
</inDoc>
```

When the action is INSERT, an attempt is made to insert the values in the following example:

```
<inDoc>
   <DBValues>
      <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
      <time_to_perform>2880</time_to_perform>
      <workflow_id>7015</workflow_id>
      <timeStamp>1062126848411</timeStamp>
      <activity>RequestPurchaseOrder</activity>
      <process_flow>send</process_flow>
   </DBValues>
</inDoc>
```

## Output from Service to Business Process

The following parameters are passed from the service to a business process when the input action is SELECT or SELECT_ID:

| Field | Description |
|-------|-------------|
| OutValues | Contains the retrieved values. Only applicable when the input action parameter is set to SELECT or SELECT_ID. |

The following output values will only be returned if a successful selection (that is, action=SELECT) is made:

```
<OutValues>
   <OutDoc>
      <result>1</result>
   </OutDoc>
</OutValues>
```

The following output values will only be returned if a successful workflow_id selection (that is, action=SELECT_ID) is made:

```
<OutValues>
   <OutDoc>
      <workflow_id>1018</ workflow_id >
   </OutDoc>
</OutValues>
```

The following parameters need to be passed from the business process to the service when it runs with the output message set to processDocument:

| Field | Description |
|-------|-------------|
| action | Indicates to the service what action needs to be performed. Required. Valid value is INSERT. |

When the action is INSERT, an attempt is made to insert the values shown below. The ctr value show below indicates the nth document. Typically, the insertion of the documents is done in a loop:

```
<inDoc>
   <DBValues>
      <trx_id>sgcentennial::4bdb20:f74ba603b7:-71f9</trx_id>
      <activity>ConfirmPurchaseOrder</activity>
      <doc_id>sgcentennial:4bdb20:f74ba603b7:-6166</doc_id>
      <ctr>1</ctr>
   </DBValues>
 </inDoc>
```

## Business Process Example

The following example assumes that the nodes DBValues are found in the process data:

```
<operation>
 <participant name="BPSSCorrelation"/>
 <output message="processTransaction">
 <assign to="conversation_id" from="conversation_id/text()"/>
 <assign to="cpa_id" from="cpa_id/text()"/>
 <assign to="action" from="'SELECT'"/>
 <assign to="inDoc" from="DBValues"/>
 </output>
 <input message="TransactionResponse">
```

```
    <assign to="DBValues/trx_id" from="OutValues/OutDoc/trx_id/text()"/>
    <assign to="DBValues/time_to_perform"
        from="OutValues/OutDoc/time_to_perform/text()"/>
   </input>
 </operation>
 <operation>
  <participant name="BPSSCorrelation"/>
  <output message="processActivity">
  <assign to="action" from="'SELECT'"/>
  <assign to="inDoc" from="DBValues"/>
  </output>
  <input message="ActivityResponse">
  <assign to="DBValues/result" from="OutValues/OutDoc/result/text()"/>
  </input>
 </operation>
 <operation>
  <participant name="BPSSCorrelation"/>
  <output message="processDocument">
  <assign to="action" from="'INSERT'"/>
  <assign to="inDoc" from="DBValues"/>
  </output>
  <input message="DocumentResponse">
  <assign to="." from="*"/>
  </input>
 </operation>
```

# Chapter 57. ebXML BPSS Doc Parsing Service

The following table provides an overview of the ebXML BPSS Doc Parsing service:

| System name | BPSSDocParser |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Validates the document based on the Conditional Expression specified in the BPSS. |
| Business usage | It helps to determine the SUCCESS or FAILURE of the state of the transaction. |
| Usage example | A business process that needs to determine the end state of the transaction can invoke this service by passing the required parameters. |
| Preconfigured? | Must be installed and deployed before it can run. There are no configuration parameters required. |
| Requires third party files? | No |
| Platform availability | All supported 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 |

## Implementing the ebXML BPSS Doc Parsing Service

To implement the ebXML BPSS Doc Parsing service, complete the following tasks:

1. Create an ebXML BPSS Doc Parsing service configuration. See *Managing Services and Adapters*.
2. Configure the ebXML BPSS Doc Parsing service. For information, see *Configuring the ebXML BPSS Doc Parsing Service*.
3. Use the ebXML BPSS Doc Parsing service in a business process.

## Configuring the ebXML BPSS Doc Parsing Service

To configure the ebXML BPSS Doc Parsing service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| documentId | The ID of the document. Required. |
| documentKey | Contains the value PrimaryDocument. |
| expression | The conditional expression as specified in the BPSS. Enclose the expression with &apos; instead of single quotation marks. Required. |

## Output from Service to Business Process

The following values are passed from the service to a business process:

| Field | Description |
|---|---|
| validityStatement/validate/value | Indicates if the documents is validated correctly. Possible values are true and false. |

The following examples show an input message coming back to the business process:

Example 1

```
<validityStatement>
    <validate>
        <value>true</value>
    </validate>
</validityStatement>
```

Example 2

```
<validityStatement>
    <validate>
        <value>true</value>
    </validate>
</validityStatement>
```

## Business Process Example

This business process assumes that the output message values are found in the process data:

```
<operation name="ValidateDoc">
 <participant name="BPSSDocParser"/>
 <output message="validateCondExpr">
 <assign to="documentId"
from="documents/document[1]/doc:document-id/text()"/>
 <assign to="expression"
    from="string(ProcessSpecification/Package/BinaryCollaboration
     [@name=//binaryCollaboration/text()]
/Success[@fromBusinessState=//businessTransactionActivity/text()]/
     ConditionExpression/@expression)"/>
 </output>
 <input message="validateCondResponse">
 <assign to="result" from="validityStatement/validate/value/text()"/>
 </input>
</operation>
```

# Chapter 58. ebXML Business Service Handler (BSI) service

The following table provides an overview of the ebXML Business Service Handler (BSI) service:

| System name | ebXMLBSIService |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B - ebXML |
| Description | Invokes the Business Service Handler to interpret and execute activities defined in the BPSS.<br>**Note:** BPSS 1.05 is supported. |
| Business usage | • Executes either inbound or outbound processing for BPSS.<br>• Determines start, finish and transition activities.<br>• Invokes appropriate services to do document schema validation. |
| Usage example | A business process that needs to execute activities specified in the BPSS. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported 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 ebXML BSI service to the business process:

| Parameter | Description |
|---|---|
| wfc | Contents of the updated business process context. Required. |

## Output from Business Process to Service

The following table describes the output from the business process to the ebXML BSI service:

| Parameter | Description |
|---|---|
| wfc | The contents of the business process context. Required. |

## Business Process Example

The following example assumes that the relevant activity_state (if needed) is set.

```
<operation>
    <participant name="ebXMLBSIService"/>
    <output message="XOut">
        <assign to="." from="*"/>
    </output>
    <input message="XIn">
        <assign to="." from="*"/>
    </input>
</operation>
```

## Output Parameters

This section contains information about calling the ebXML service for outbound and inbound activities.

## Output Parameters – Outbound Processing

In processing outbound activities, there are two possible ways of calling the ebXML BSI service. The first way, with the activity_flag set to pre indicates preprocessing of activities before the internal process runs. The second way, with the activity_flag set to post, indicates postprocessing of activities after the internal process runs.

```
<assign to="activity_state" from="'pre'"/>
<operation name="One">
    <participant name="ebXMLBSIService"/>
    <output message="Xout">
        <assign to="." from="*"/>
    </output>
    <input message="Xin">
        <assign to="." from="*"/>
    </input>
</operation>
<assign to="activity_state" from="'post'"/>
<operation name="One">
    <participant name="ebXMLBSIService"/>
    <output message="Xout">
        <assign to="." from="*"/>
    </output>
    <input message="Xin">
        <assign to="." from="*"/>
    </input>
</operation>
```

## Output Parameters – Inbound Processing

When processing inbound activities, just call the service, as shown in the following example:

```
<operation name="One">
   <participant name="ebXMLBSIService"/>
   <output message="Xout">
      <assign to="." from="*"/>
   </output>
   <input message="Xin">
        <assign to="." from="*"/>
   </input>
</operation>
```

## Implementing the ebXML BSI Service

To implement the ebXML BSI service for use in a business process:

1. Create an ebXML BSI service configuration. See *Managing Services and Adapters*.
2. Use the ebXML BSI service in a business process.

# Chapter 59. ebXML CPA Lookup Service

The following table provides an overview of the ebXML CPA Lookup service:

| System name | ebXMLCPALookupService |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B - ebXML |
| Description | Retrieve CPA information based on service/action or business request.<br>**Note:** CPA 2.0 is supported. |
| Business usage | Used for retrieving CPA information before creating a new ebXML outbound message or before validating an ebXML inbound message. |
| Usage example | A business process that needs to retrieve specific CPA information can invoke this service by passing the required parameters. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Restrictions | No |
| Persistence level | System default |
| Testing considerations | To test a configuration of the ebXML CPA Lookup service, provide the the necessary input parameters. |

## Output from Service to Business Process

The following table describes the output from the ebXML CPA Lookup service to the business process:

| Parameter | Description |
|---|---|
| cpa | CPA info |

## Output from Business Process to Service

The following table describes the output from the business process to the ebXML CPA Lookup service:

| Parameter | Description |
|---|---|
| cpa_id | A unique value that identifies the alias name of CPA. Required. |

| Parameter | Description |
|---|---|
| service | A unique value that identifies the service of the ebXML message |
| serviceType | A unique value that identifies the service type of the ebXML message. Optional. |
| action | A unique value that identifies the action of the ebXML message. Optional. |
| b2b-message-mode | A flag for ebXML Lite which is passed from trading partner. Valid values are Send and Respond. Optional. |
| BPSSMode | A flag that identifies whether ebXML message is outbound or inbound. Valid values are Send and Receive. Required. |
| thisPartyId | A unique value that identifies the party name or ID of the current Trading Partner. Optional. |
| fromb2bInbound | A flag that identifies that the message is not the first request. Valid values are "" or true. Optional. |
| binaryCollaboration | A unique value that identifies the binaryCollaboration of the ebXML message. Optional. |
| businessTransactionActivity | A unique value that identifies the businessTransactionActivity of the ebXML message. Optional. |
| requestOrResponseAction | A unique value that identifies the requestOrResponseAction of the ebXML message. Optional. |

## Business Process Example

The following example illustrates using the ebXML CPA Lookup service in a business process:

```
<process name="testEBXMLCPALookupService">
    <sequence>
        <operation name="ebXMLCPALookup">
            <participant name="ebXMLCPALookupService"/>
            <output message="ebXMLCPALookupInputMessage">
                <assign to="." from="*"/>
            </output>
            <input message="ebXMLCPALookupOutputMessage">
                <assign to="." from="*"/>
            </input>
        </operation>
    </sequence>
</process>
```

## Implementing the ebXML CPA Lookup Service

To implement the ebXML CPA Lookup service for use in a business process:

1. Create an ebXML CPA Lookup service configuration. See *Managing Services and Adapters*.
2. Use the ebXML CPA Lookup service in a business process.

# Chapter 60. ebXML Lookup Service

The following table provides an overview of the ebXML Lookup service:

| System name | ebXMLLookupService |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Retrieves the content of either CPA or BPSS schema, based on the alias name of the schema and the type of schema. |
| Business usage | Used for the ebXML Messaging service and the BPSS Handler service to retrieve the schema information of CPA or BPSS. |
| Usage example | A business process that needs to retrieve CPA or BPSS schema can invoke this service by passing the required parameters. |
| Preconfigured? | No. Must be installed and deployed before it can run. There are no configuration parameters required. |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Restrictions | No |
| Testing considerations | To test a configuration of the ebXML Lookup service, make sure you have the CPA or BPSS schema created under Deployment/EBXML/BPSS or CPA. |

## How the ebXML Lookup Service Works

The ebXML Lookup service retrieves the content of either CPA or BPSS schema, based on the alias name of schema and the type of schema.

## Parameters Passed from Business Process to Service

The following parameters are passed to the service when it runs with the output message set to lookupEBXML:

| Field | Description |
|---|---|
| ebxmlName | A unique value that identifies the alias name of CPA or BPSS schema. |
| schemaType | A value that identifies the type of schema to be retrieved. Possible values are CPA and BPSS. |

## Business Process Example

The following example shows how the ebXML Lookup service might be used in a business process:

```
Retrieve CPA schema:
<operation>
 <participant name="ebXMLLookupService"/>
 <output message="lookupEBXML">
  <assign to="ebxmlName" from="cpaId/text()"/>
  <assign to="schemaType" from="'CPA'"/>
 </output>
 <input message="getEBXML">
  <assign to="cpa-store" from="ebxmlSchema/node()"/>
 </input>
</operation>
Retrieve BPSS schema:
<operation>
<participant name="ebXMLLookupService"/>
 <output message="lookupEBXML">
  <assign to="ebxmlName" from="bpssName/text()"/>
  <assign to="schemaType" from="'BPSS'"/>
 </output>
 <input message="getEBXML">
  <assign to="." from="ebxmlSchema/node()"/>
 </input>
</operation>
```

## Implementing the ebXML Lookup Service

To implement the ebXML Lookup service, complete the following tasks:

1. Activate your license for the ebXML Lookup service. See *Installing Sterling Integrator*.
2. Create a ebXML Lookup service configuration. See *Managing Services and Adapters*.
3. Configure the ebXML Lookup service. See *Configuring the ebXML Lookup Service*.
4. Use the ebXML Lookup service in a business process.

## Configuring the ebXML Lookup Service

To configure the ebXML Lookup service, you must specify field settings in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| ebXMLName | Unique value that identifies the alias name of CPA or BPSS schema. |
| schemaType | A value that identifies the type of schema to be retrieved. Possible values are CPA and BPSS. |

# Chapter 61. ebXML Manifest and Payload Service

The following table provides an overview of the ebXML Manifest and Payload service:

| System name | ManifestPayloadService |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Create eb:Manifest and payloads node before constructing ebXML outbound message. |
| Business usage | Used for creating eb:Manifest and payloads node based on documents/document. |
| Usage example | A business process that needs to create eb:Manifeset and payloads with documents/document input structure can invoke this service by passing the required parameters. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Restrictions | No |
| Persistence level | System default |
| Testing considerations | To test a configuration of the Manifest and Payload service, provide the documents/document structure. |

## Output from Service to Business Process

The following table describes the output from the ebXML Manifest and Payload service to the business process:

| Parameter | Description |
|---|---|
| eb:Manifest | Manifest information prepared for SOAP-ENV:Body. |
| payloads | Payload information prepared for mime:message/mime:body |

## Output from Business Process to Service

The following table describes the output from the business process to the ebXML Manifest and Payload service:

| Parameter | Description |
|---|---|
| documents | A node contains the document information. Required. |

## Business Process Example

The following example illustrates using the ebXML Manifest and Payload service in a business process:

```
<process name="testManifestandPayloadService">
    <sequence>
    <operation name="CreateManifestPayload">
        <participant name="ManifestPayloadService"/>
        <output message="ManifestPayloadInputMessage">
            <assign to="outboundDoc" from="documents/node()"/>
        </output>
        <input message="ManifestPayloadOutputMessage">
            <assign to="." from="Manifest/node()"/>
            <assign to="." from="Payload/node()"/>
        </input>
    </operation>
    </sequence>
</process>
```

## Implementing the ebXML Manifest and Payload Service

To implement the ebXML Manifest and Payload service for use in a business process:

1. Create an ebXML Manifest and Payload service configuration. See *Managing Services and Adapters*.
2. Use the ebXML Manifest and Payload service in a business process.

# Chapter 62. ebXML Message Handler Service

The following table provides an overview of the ebXML Message Handler service:

| System name | ebXML Message Handler service |
|---|---|
| Graphical Process Modeler (GPM) categories | Internet B2B > ebXML |
| Description | Composes outgoing and decomposes incoming ebXML messages based on CPA and BPSS parameters. |
| Business usage | For an outbound message the service is used to pack the required message in the specified format before sending it out. For an inbound message the service is used to unpack and validate MIME and SOAP structure of the message before passing it on. |
| Usage example | A business process that needs to compose or unpack ebXML message can invoke this service to handle both outbound and inbound ebXML processes. For more details see *Parameters Passed from a Business Process*. |
| Preconfigured? | No. This service must be installed and deployed before it can be invoked. There are no configuration parameters required. |
| 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 | **SUCCESS** |
| --- | --- |
| | ebXML message is composed or decomposed successfully |
| | **SUCCESS** |
| | The original Acknowledgement of this request has been sent out. |
| | **ERROR** |
| | Failed to create ebXML MIME message structure |
| | **ERROR** |
| | Failed to digitally sign ebXML message |
| | **ERROR** |
| | Failed to encrypt ebXML message |
| | **ERROR** |
| | Failed to MIME ebXML message |
| | **ERROR** |
| | Could not find private cert in CPA before digitally signing the ebXML message |
| | **ERROR** |
| | Could not find encryption cert in CPA before encrypting the ebXML message |
| | **ERROR** |
| | Could not find public key from database |
| | **ERROR** |
| | There is no support transport protocol agreed in CPA |
| | **ERROR** |
| | There is no endpoint agreed in CPA |
| | **ERROR** |
| | Failed to crack ebXML inbound MIME message |
| | **ERROR** |
| | The inbound MIME message is not conforming to the SOAP structure |
| | **ERROR** |
| | The inbound MIME message does not have an end boundary |
| | **ERROR** |
| | There is no transport information to send original Acknowledgement |
| Restrictions | None |

## Business Process Example

The following example assumes that the nodes DBValues are found in the process data:

```
<process name="testEBXMLMSHService">
   <sequence>
      <operation>
         <participant name="ebXMLMSHService"/>
         <output message="ebXMLMSHInputMessage">
            <assign to="." from="*"/>
         </output>
         <input message="ebXMLMSHOutputMessage">
            <assign to="." from="*"/>
```

```
                </input>
            </operation>
        </sequence>
    </process>
```

# Chapter 63. ebXML Profile Service

The ebXML Profile service returns a specified business process name to be run. The service is used to reuse a business process and to prevent more than one instance of the same business process from being started. The following table provides an overview of the ebXML Profile service:

| System name | ebXMLProfile |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Returns the name of a business process to be run. |
| Business usage | Used in conjunction with ebXML messaging. It determines the business process to be run based on the CPA ID, service and action values passed as part of the ebXML message. |
| Usage example | A business process that correctly validates an ebXML message needs information about what other business process needs to be run in response to the ebXML message. |
| Preconfigured? | Must be installed and deployed before it can run. There are no configuration parameters required. |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Returned status values | None |
| Restrictions | No |
| Notes | The output message is a node containing the business process to be run.<br><br>`<callback>`<br>`<messageName>DoIt</messageName>`<br>`<processDefinition>`<br>`<workFlowName>TestServerIn</workFlowName>`<br>`</processDefinition>`<br>`</callback>` |

## Parameters Passed from Business Process to Service

The following parameters need to be passed to the service when it runs with the output message set to getCallbackRequest.

| Field | Description |
|---|---|
| CPAId | Unique value that identifies a trading partner relationship as defined in the CPA specification. |
| Service | Identifies a set of business transactions as defined in the CPA specification. |

| Field | Description |
|---|---|
| Action | Identifies a business transaction as defined in the CPA specification. |

### Business Process Example

The following example shows how the ebXML Profile service might be used in a business process:

```
<operation>
 <participant name="ebXMLProfile"/>
 <output message="getCallbackRequest">
 <assign to="CPAId" from="InboundMIME/mime:message/mime:body/mime:message[1]
      /mime:body/SOAP-ENV:Envelope/SOAP-ENV:Header/eb:
      MessageHeader/eb:CPAId/text()"/>
 <assign to="Service" from="InboundMIME/mime:message/mime:body/mime:
      message[1]/mime:body/SOAP-ENV:Envelope/SOAP-ENV:Header/eb:
      MessageHeader/eb:Service/text()"/>
 <assign to="Action" from="InboundMIME/mime:message/mime:body/mime:message[1]
      /mime:body/SOAP-ENV:Envelope/SOAP-ENV:Header/eb:MessageHeader/
      eb:Action/text()"/>
 </output>
 <input message="getCallbackResponse">
 <assign to="Client" from="callback/node()"/>
 </input>
</operation>
```

### Implementing the ebXML Profile Service

To implement the ebXML Profile service, complete the following tasks:

1. Activate your license for the ebXML Profile service. See *Installing Sterling Integrator*.
2. Create an ebXML Profile service configuration. See *Managing Services and Adapters*.
3. Use the ebXML Profile service in a business process.

# Chapter 64. ebXML Request Response Service

The ebXML Request Response service generates unique conversation IDs and message IDs that are used in an ebXML conversation. The following table provides an overview of the ebXML Request Response service:

| | |
|---|---|
| System name | ebXMLRequestResponse |
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Returns the name of a business process to be run. |
| Business usage | Used with ebXML messaging. Creates unique values of conversation and message IDs. |
| Usage example | A business process that is creating an ebXML message needs to generate a dialog or message ID if the current message is part of a group of messages, which should be treated as part of the same dialog. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Returned status values | None |
| Restrictions | None |

## Parameters Passed from Business Process to Service

The following parameters need to be passed to the service when it runs with the output message set to asssociateRequest.

| Field | Description |
|---|---|
| PartyId | Unique value that identifies a trading partner relationship as defined in the CPA specification. Required. |
| ConversationId | Identifies a set of business transactions as defined in the CPA specification. |
| MessageId | Identifies a business transaction as defined in the CPA specification. |
| handler | Handler, that is, the ID of the ebXML Request Response service. Numeric. Valid value is a system-generated ID. |

## Business Process Example

The following is an example of a business process that uses the ebXML Request Response service:

```
<operation>
 <participant name="ebXMLRequestResponse"/>
<output message="associateRequest">
 <assign to="PartyId" from="string(cpa/counterParty/eb:PartyId)"/>
 <assign to="handler" from="thisProcessInstance/node()"/>
</output>
<input message="associateResponse">
 <assign to="conversationId" from="ConversationId/node()"/>
 <assign to="messageId" from="MessageId/node()"/>
</input>
</operation>
```

## Implementing the ebXML Request Response Service

To implement the ebXML Request Response service, complete the following tasks:

1. Activate your license for the ebXML Request Response service. See *Installing Sterling Integrator*.
2. Create an ebXML Request Response service configuration. See *Managing Services and Adapters*.
3. Use the ebXML Request Response service in a business process.

# Chapter 65. ebXML Validation Service

The following table provides an overview of the ebXML Validation service:

| System name | ebXMLValidationService |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Validates inbound ebXML message to make sure it conforms to the ebXML structure and verify the validity of the digital signature. If there is a message level exception found, returns error list. Supported ebXML stacks are:<br>**Note:** ebMS 2.0 is supported. |
| Business usage | Used for ebXML Messaging to validate inbound message level content. |
| Usage example | A business process that needs to validate ebXML inbound message can invoke this service by passing the required parameters. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported 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 | None |
| Restrictions | No |
| Persistence level | System default |
| Testing considerations | None |

## Output from Service to Business Process

The following table describes the output from the ebXML Validation service to the business process:

| Parameter | Description |
|---|---|
| errors | Error List. Valid value is eb:ErrorList or eb:Error. Optional. |

## Output from Business Process to Service

The following table describes the output from the business process to the ebXML Validation service:

| Parameter | Description |
|---|---|
| inboundMIME | Node contains the inbound ebXML message. Required. |

| Parameter | Description |
|---|---|
| CPA | Node contains the CPA info based on the inbound ebXML message service/action. Required. |
| BPSSParam | Node contains the BPSS parameters information based on the return of BPSSHandler. Optional. |
| InboundDocId | Node contains the document-id of the inbound ebXML message. Required. |

## Business Process Example

The following example illustrates using the ebXML Validation service in a business process:

```
<process name="testEBXMLValidationService">
    <sequence>
      <operation name="ebXMLValidation">
        <participant name="ebXMLValidationService"/>
          <output message="ebXMLValidationInputMessage">
            <assign to="inboundMIME" from="InboundMIME/node()"/>
            <assign to="CPA" from="cpa/node()"/>
            <assign to="BPSSParam" from="bpssParams/node()"/>
            <assign to="InboundDocId"
              from="string(inbound-mime-document/doc:document-id)"/>
          </output>
          <input message="ebXMLValidationOutputMessage">
            <assign to="." from="errors"/>
          </input>
      </operation>
    </sequence>
</process>
```

## Implementing the ebXML Validation Service

To implement the ebXML Validation service for use in a business process:

1. Create an ebXML Validation service configuration. See *Managing Services and Adapters*.
2. Use the ebXML Validation service in a business process.

# Chapter 66. ebXML XML Digital Signature Service

The ebXML XML Digital Signature service composes and verifies digital signatures. The following table provides an overview of the ebXML XML Digital Signature service:

| | |
|---|---|
| System name | ebXMLDSig |
| Graphical Process Modeler (GPM) categories | All Services, Internet B2B > ebXML |
| Description | Composes and verifies digital signatures. |
| Business usage | Business usage:<br>• Composes a digital signature (XMLDSIG) on a given node or primary document by using a private key of a specified and returns a hash value together with the original input node.<br>• Validates the hash value of the input node or Primary Document by making use of a public key, either through a specified certificate or from the KeyInfo element of the Signature and indicates this with a TRUE or FALSE. |
| Usage example | A business process that needs a node to be digitally signed or verified can invoke this service by passing the required parameters. |
| Preconfigured? | Must be installed and deployed before it can run. There are no configuration parameters required. |
| Requires third party files? | xss4j.jar (included in Sterling Integrator installation) |
| 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 | None |
| Returned status values | None |
| Restrictions | None |
| Testing considerations | Use the correct certificates for the signing. The most frequent problem encountered is the certificates used for the signing were not created with a storepass value of `integrator` and a keypass value of `integrator`. If you receive an error with this condition, see your system administrator. |

| | |
|---|---|
| Notes | Output Parameters: signRequestA node or updated Primary Document containing the original contents and the Signature element after the digital signature is applied. Output Parameters: verifyRequestA node containing the validity of the signature. The two possible values are:<br><br>```<br><validity><br><valid>true</value><br></validity><br>```<br><br>and<br><br>```<br><validity><br><valid>false</value><br></validity><br>An updated Primary Document when validating<br>the Primary Document<br>``` |

## Parameters Passed from Business Process to Service

The following table describes the parameters that are passed from a business process to the service:

| Field | Description |
|---|---|
| certificateIdentifier | Alias of a certificate public key. When used during signing, it indicates that the KeyInfo element must be included in the Signature. Only applicable when signing the Primary Document. Valid value is a valid alias name. Required. |
| ds:Transforms | Required Transforms to be used in the signing. If omitted, the enveloped-signature Transform will be used. Only applicable when signing the Primary Document. Valid value is a valid node. |
| incomingDoc | Node to be digitally signed. When not specified, it implies that the Primary Document needs to be signed. Valid value is a valid node. |
| nodeToSign | Required node to be signed. This node is found within the Primary Document. If not specified it implies that we wish to sign the whole document. Only applicable when signing the Primary Document. Valid value is a valid node name. |
| signCertificateIdentifier | Alias of a private key of a certificate. Valid value is a valid alias name. |

The incoming document must contain the prerequisites of the ebXML XML Digital Signature service. The following is an example node that is passed to the ebXML XML Digital Signature service for signature creation:

```
<ebXMLMessage>
<mime:message xmlns:mime="http://www.company.com/mime/v0.5">
<mime:header name="Content-Type">multipart/related
<mime:parameter name="start">ebxml-envelope@company.com</mime:parameter>
<mime:parameter name="type">text/xml</mime:parameter>
</mime:header>
<mime:header name="SOAPAction">ebXML</mime:header>
<mime:body>
<mime:message>
<mime:header name="Content-ID"><ebxml-envelope@company.com>
```

```
        </mime:header>
<mime:header name="Content-Type">text/xml<mime:parameter
     name="charset">UTF-8</mime:parameter>
</mime:header>
<mime:body>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="
http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
     http://www.oasis-open.org/committees/ebxml-msg/schema/envelope.xsd">
<SOAP-ENV:Header xmlns:eb=
     "http://www.oasis-open.org/committees/ebxml-msg/schema/
msg-header-2_0.xsd">
<eb:MessageHeader SOAP-ENV:mustUnderstand="1"
        eb:id="ebxml-MessageHeader-company.com"
eb:version="2.0" xsi:schemaLocation=
        "http://www.oasis-open.org/committees/ebxml-msg/
schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
<eb:From>
<eb:Role>http://www.company.com/roles/Sender</eb:Role>
</eb:From>
<eb:To>
<eb:Role>http://www.company.com/roles/Receiver</eb:Role>
</eb:To>
<eb:CPAId>CompanyID-CompanyID</eb:CPAId>
<eb:ConversationId>server::111z1:zzz999z9z:-1111</eb:ConversationId>
<eb:Service eb:type="string">FileTransfer-Sign</eb:Service>
<eb:Action>Receive</eb:Action>
<eb:MessageData>
<eb:MessageId>server::111z1:zzz999z9z:-1111</eb:MessageId>
<eb:Timestamp>2005-07-18T04:10:18Z</eb:Timestamp>
</eb:MessageData>
<eb:Description xml:lang="en-US">An ebXML Message.</eb:Description>
</eb:MessageHeader>
<eb:SyncReply SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
     SOAP-ENV:mustUnderstand="1"
eb:id="ebxml-SyncReply-company.com" eb:version="2.0" xsi:schemaLocation=
     "http://www.oasis-open.org/committees/ebxml-msg/
schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
<eb:AckRequested SOAP-ENV:mustUnderstand="1"
     eb:id="ebxml-AckRequested-company.com"
eb:signed="true" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/
     schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
</SOAP-ENV:Header>
<SOAP-ENV:Body/>
</SOAP-ENV:Envelope>
</mime:body>
</mime:message>
<mime:message>
<mime:header name="Content-ID"><ebxml-payload-0@company.com></mime:header>
<mime:header name="Content-Type">application/xml</mime:header>
<mime:body>
<doc:document-id xmlns:doc="http://www.company.com/document-id">
     server:4llll4kdlllrrrr4:-99zz</doc:document-id>
</mime:body>
</mime:message>
</mime:body>
</mime:message>
</ebXMLMessage>
</mime:message>
</mime:body>
</mime:message>
</ebXMLMessage>
```

The following parameters need to be passed to the service when it runs with the output message set to verifyRequest:

| Field | Description |
| --- | --- |
| certificateIdentifier | Alias of a public key of a certificate. Valid value is a valid alias name. |
| documentID | Document-id of the MIME message to be cerified. Valid value is a valid document id. |
| incomingDoc | Node that needs to have its digital signature verified. Valid value is a valid node. |
| removeSignature | Indicates if the validated document needs to have the Signature element removed. Only applicable when validating the Primary Document. Valid values are True and False. |

The incoming document must contain the prerequisites of the ebXML XML Digital Signature service. The following is an example node that is passed to the ebXML XML Digital Signature service for signature validation:

```
<ebXMLMessage>
 <mime:message xmlns:mime="http://www.company.com/mime/v0.5">
 <mime:header name="Content-Type">multipart/related<mime:
        parameter name="start">ebxml-envelope@company.com</mime:parameter>
<mime:parameter name="type">text/xml</mime:parameter>
 </mime:header>
 <mime:header name="SOAPAction">ebXML</mime:header>
 <mime:body>
  <mime:message>
    <mime:header name="Content-ID"><ebxml-envelope@company.com>
</mime:header>
    <mime:header name="Content-Type">text/xml<mime:
                parameter name="charset">UTF-8</mime:parameter>
    </mime:header>
 <mime:body>
 <SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/
soap/envelope/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.xmlsoap.org/soap/envelope/
        http://www.oasis-open.org/committees/ebxml-msg/schema/
envelope.xsd">
 <SOAP-ENV:Header xmlns:eb="http://www.oasis-open.org/committees/
        ebxml-msg/schema/msg-header-2_0.xsd">
 <eb:MessageHeader SOAP-ENV:mustUnderstand="1" eb:
        id="ebxml-MessageHeader-company.com"
eb:version="2.0" xsi:schemaLocation="http://www.oasis-open.org/committees/
        ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd">
 <eb:From>
 <eb:Role>http://www.company.com/roles/Sender</eb:Role>
 </eb:From>
 <eb:To>
 <eb:Role>http://www.company.com/roles/Receiver</eb:Role>
 </eb:To>
 <eb:CPAId>company-company</eb:CPAId>
 <eb:ConversationId>server::11c1:88888zzzz:-1111</eb:ConversationId>
 <eb:Service eb:type="string">FileTransfer-Sign</eb:Service>
 <eb:Action>Receive</eb:Action>
 <eb:MessageData>
 <eb:MessageId>server::11c1:88888zzzz:-1111</eb:MessageId>
 <eb:Timestamp>2002-07-18T04:10:18Z</eb:Timestamp>
 </eb:MessageData>
 <eb:Description xml:lang="en-US">An ebXML Message.</eb:Description>
```

```
</eb:MessageHeader>
<eb:SyncReply SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"
      SOAP-ENV:mustUnderstand="1"
eb:id="ebxml-SyncReply-company.com" eb:version="2.0"
      xsi:schemaLocation="http://www.oasis-open.org/committees/
      ebxml-msg/schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
<eb:AckRequested SOAP-ENV:mustUnderstand="1"
      eb:id="ebxml-AckRequested-company.com"
eb:signed="true" eb:version="2.0"
xsi:schemaLocation="http://www.oasis-open.org/committees/ebxml-msg/
      schema/msg-header-2_0.xsd
http://www.oasis-open.org/committees/ebxml-msg/schema/msg-header-2_0.xsd"/>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm=
      "http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
<ds:Reference Type="http://www.w3.org/2000/09/xmldsig#Object" URI="">
<ds:Transforms>
<ds:Transform Algorithm="
http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
<ds:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
<ds:XPath> not(ancestor-or-self::node()[@SOAP-ENV:actor=
      "urn:oasis:names:tc:ebxml-msg:service:nextMSH"]|ancestor-
or-self::node()
      [@SOAP-ENV:actor="http://schemas.xmlsoap.org/soap/actor/next"])
          </ds:XPath> </ds:Transform>
<ds:Transform Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
<ds:DigestValue>+TTgggfFZZZ+444t444DDffEEEddddd=</ds:DigestValue>
</ds:Reference>
<ds:Reference URI="cid:ebxml-payload-0@company.com">
          <ds:DigestMethod Algorithm="http://www.w3.org/
2000/09/xmldsig#sha1"/>
<ds:DigestValue>5SS4d44dGG1DD1DDddd3FFFee2GGGddd=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>
ZZzzzZZzzzzZZzzzZZzzzzZZZzAAaaaEEaaaEEEEaaaaEEEaaaaDDDDddDDdddzzzZzzZZzzzZ
zzAA33AAzzz44ZZaaZZZaaZZZaaaZZZZ4eeEEE5WWWqqqEEEaaa4AAAee5DDDddEEEEddd2DDD
9Ik99R3EeeR444rrrFFF4694eeeOOO333aaddii999l=
  </ds:SignatureValue>
</ds:Signature>
</SOAP-ENV:Header>
<SOAP-ENV:Body/>
</SOAP-ENV:Envelope>
</mime:body>
</mime:message>
<mime:message>
<mime:header name="Content-ID"><ebxml-payload-0@company.com></mime:header>
<mime:header name="Content-Type">application/xml</mime:header>
<mime:body>
<doc:document-id xmlns:doc="http://www.company.com/document-id">
    server:999z9:ggggh9g9g:-99zz</doc:document-id>
</mime:body>
</mime:message>
</mime:body>
</mime:message>
</ebXMLMessage>
```

## Implementing the ebXML XML Digital Signature Service

To implement the ebXML XML Digital Signature service, complete the following tasks:

1. Activate your license for the ebXML XML Digital Signature service. See *Installing Sterling Integrator*.
2. Create an ebXML XML Digital Signature service configuration. See *Managing Services and Adapters*.
3. Configure the ebXML XML Digital Signature service. For information, see *Configuring the ebXML XML Digital Signature Service*.
4. Use the ebXML XML Digital Signature service in a business process.

   Import a Key certificate for the ebXML XML Digital Signature service to reference in the business process. For the procedure, see *Importing a Key Certificate into a Business Process*.

## Configuring the ebXML XML Digital Signature Service

To configure the ebXML XML Digital Signature service, you must specify settings for the following fields in the GPM:

| Field | Description |
| --- | --- |
| Config | Name of the service configuration. |
| certificateIdentifier | The alias of a public key of a certificate. Valid value is a valid alias name. |
| incomingDoc | The node that needs to be digitally signed. Valid value is a valid node. |
| signCertificateIdentifier | The alias of a private key of a certificate. Valid value is a valid alias name. |

## Importing a Key Certificate into a Business Process

When importing a Key certificate, your BPML should use this alias to associate with the appropriate field (signCertificateIdentifier). The following BPML provides an example of how your BPML should look:

```
<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="incomingDoc" from="ebXMLMessage/mime:message"/>
<assign to="signCertificateIdentifier" from="cpa/thisParty/
        signing-certificate-name/text()"/>
</output>
<input message="signResponse">
<assign to="ebXMLMessage"
from="signedEnvelope/signRequest/incomingDoc/node()"/>
</input>
</operation>
```

## Business Process Example

The following example assumes that the nodes are found in the process data:

```
<process name="Test_Sign>
<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="incomingDoc" from="ebXMLMessage/mime:message"/>
<assign to="signCertificateIdentifier" from="cpa/thisParty/signing-
        certificate-name/text()"/>
</output>
<input message="signResponse">
<assign to="ebXMLMessage"
```

```
from="signedEnvelope/signRequest/incomingDoc/node()"/>
</input>
</operation>
</process>
```

The following example shows how to sign the whole Primary Document and add the KeyInfo element with the Signature element by including the certificateIdentifier parameter in the signing request. The example also includes how to construct the ds:Transforms node:

```
<process name="TestSigning">
<sequence>
<assign to="temp/@Algorithm" from="'http://www.w3.org/2000/09/
    xmldsig#enveloped-signature'"/>
<assign to="ds:Transforms/ds:Transform" from="temp/@*"/>
<assign to="temp/@Algorithm" from="'http://www.w3.org/TR/1999/
    REC-xpath-19991116'"/>
<assign to="temp/ds:XPath">count(/ETrade/DataArea/LCAdvice/descendant-
    or-self::node()
    | /ETrade/DataArea/LCAdvice/descendant-or-self::*/@* |
     /ETrade/DataArea/LCAdvice/descendant-or-self::*/namespace::*)
    = count(. | /ETrade/DataArea/LCAdvice/descendant-or-self::node() |
    /ETrade/DataArea/LCAdvice/descendant-or-self::*/@* |
    /ETrade/DataArea/LCAdvice/descendant-or-self::*/namespace::*)
</assign>
<assign to="ds:Transforms/ds:Transform" from="temp/@* | temp/node()"
    append="true"/>
<!-- Do the Signing -->
<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="." from="*"/>
<assign to="signCertificateIdentifier" from="'PrivKey'"/>
<assign to="certificateIdentifier" from="'PubKey'"/>
<assign to="ds:Transforms" from="ds:Transforms/node()"/>
</output>
<input message="signResponse">
<assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

This BPML example shows how to sign a particular node in the Primary Document

```
<process name="TestSigning">
<sequence>
<assign to="temp/@Algorithm" from="'http://www.w3.org/2000/09/
    xmldsig#enveloped-signature'"/>
<assign to="ds:Transforms/ds:Transform" from="temp/@*"/>
<assign to="temp/@Algorithm" from="'http://www.w3.org/TR/1999/
    REC-xpath-19991116'"/>
<assign to="temp/ds:XPath">count(/ETrade/DataArea/LCAdvice/descendant
    -or-self::node()
    | /ETrade/DataArea/LCAdvice/descendant-or-self::*/@* |
    /ETrade/DataArea/LCAdvice/descendant-or-self::*/namespace::*)
    = count(. | /ETrade/DataArea/LCAdvice/descendant-or-self::node() |
    /ETrade/DataArea/LCAdvice/descendant-or-self::*/@* |
    /ETrade/DataArea/LCAdvice/descendant-or-self::*/namespace::*)
</assign>
<assign to="ds:Transforms/ds:Transform" from="temp/@* | temp/node()"
    append="true"/>
<!-- Do the Signing -->
<operation name="SignMessage">
<participant name="ebXMLDSig"/>
<output message="signRequest">
<assign to="." from="*"/>
```

```
<assign to="signCertificateIdentifier" from="'PrivKey'"/>
<assign to="nodeToSign" from="'TransactionResultResponse'"/>
<assign to="certificateIdentifier" from="'PubKey'"/>
<assign to="ds:Transforms" from="ds:Transforms/node()"/>
</output>
<input message="signResponse">
<assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

This BPML example shows how to validate a signed Primary Document using a specified public certificate stored in GIS. The signature element will be removed from the Primary Document after the successful validation.

```
<process name="TestValidating">
<sequence>
<!-- Do the verification -->
<operation name="VerifyMessage">
<participant name="ebXMLDSig"/>
<output message="verifyRequest">
<assign to="." from="*"/>
<assign to="removeSignature" from="'true'"/>
<assign to="certificateIdentifier" from="'PubKey'"/>
</output>
<input message="verifyResponse">
<assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

# Chapter 67. EJB Adapter

The Enterprise Java Bean (EJB) adapter is comprised of the EJB Listener service and the GIS Adapter EJB (a J2EE application) that work together to enable you to launch a Sterling Integrator business process from a remote J2EE application server.

For this document a remote application server means one that is not associated with Sterling Integrator. The remote application server may reside on the same host as Sterling Integrator or not.

The following table provides an overview of the EJB Adapter Listener service:

| System name | GISEJBAdapter |
| --- | --- |
| Graphical Process Modeler (GPM) category | This service is not used in business processes. |
| Description | The EJB Adapter, using the EJB Adapter Listener service, defines a port on the Sterling Integrator host and listens for incoming requests from a remote application server. Based on the request, the service can:<br>• Initiate (bootstrap) a business process<br>• Pass a document and other data from the application server into process data of the initiated business process<br>• Return a status about the business process<br>• Return the results of the business process<br>The GIS Adapter EJB is deployed on a remote application server and is used to send requests to Sterling Integrator. |
| Business usage | The EJB Adapter enables you to integrate programs on your existing application server with the business process transaction engine provided by Sterling Integrator. |
| Usage example | You create purchase orders on your legacy application server and you want to use Sterling Integrator to route the order for approval, translate the order into EDI, transfer the order to the vendor and update the legacy application with the status of the purchase order throughout that process. Use the GIS Adapter EJB to write a client program on your application server that sends the purchase order to Sterling Integrator, initiates a business process and requests status. |
| Preconfigured? | No |
| Requires third party files? | Yes. One of the following J2EE application servers:<br>• BEA® WebLogic® 7.0 SP2 & SP3<br>• JBoss™ 3.2<br>• IBM® WebSphere® 5.0.2 |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | None |

| Application requirements | The following statements apply: |
|---|---|
| | • The Java Archive file GISInvokerEJB_*appServProductName*.JAR must be installed and configured on a WebLogic, WebSphere, or JBoss application server. |
| | See *Deploying GIS Adapter EJB on a WebSphere Application Server*, *Deploying GIS Adapter EJB on a WebLogic Application Server*, or *Deploying GIS Adapter EJB on a JBoss Application Server*. |
| | • You must create the client program on the application server. The client program is responsible for sending the requests to Sterling Integrator. |
| Initiates business processes? | Yes. This service initiates a business process on demand from a remote application server. |
| Invocation | This service is inbound (bootstrapping) only, and will launch a business process on demand (non-polled). |
| Business process context considerations | This service creates an initial business process context (WFC) with the documents and parameters provided by a client program at execution time. |
| Returned status values | If the application server needs to obtain the status or the results of the Sterling Integrator business process, you can use the following two methods in the client program: getStatus() and getResults(). |
| | • Success – The EJB call to Sterling Integrator was successful. |
| | • Error – The EJB call to Sterling Integrator was not successful. |
| Restrictions | These restrictions apply: |
| | • The communication from the application server to Sterling Integrator is asynchronous only. |
| | • The Adapter Listener service cannot be used in a business process. It is only used for initiating a business process. |
| Persistence level | Full Persistence |
| Testing considerations | To test, do the following:Install and configure the EJB Adapter Listener serviceDeploy the GIS Adapter EJB in the remote application serverWrite a client program to send requests to Sterling Integrator. See *EJB Client Program Example*. |

## How the EJB Adapter Works

The following steps illustrate a synchronous communication between the remote application server and Sterling Integrator:

1. A legacy system running on a remote J2EE application server has a document that requires processing by Sterling Integrator.

2. A client program on the remote application server uses the GIS Adapter EJB to transfer the document and request that a specific business process be started. The program can also send requests for status, and, when the status indicates complete, a request for the resulting processed document.

3. The EJB Adapter Listener service receives the request, initiates the requested business process and passes the document into process data.

4. When the service receives the status request, it passes the business process status back to the remote application server. When the service receives a result request it passes the primary document back to the remote application server.

## Implementing Communications with a Remote Application Server

To implement synchronous communication from a remote application server to Sterling Integrator two components must be set up:

- The EJB Adapter Listener service that defines a port on the Sterling Integrator host.
- The J2EE application, GISInvokeEJB_*appServProductName*.jar, that you install on the remote application server.

Complete the following process:

1. Create an EJB Adapter Listener service configuration.
2. Configure the EJB Adapter Listener service.
3. Create and enable a business process that you want to have initiated by the remote application server. For example, create a business process that translates documents from a legacy system on the remote application server.
4. Deploy the GIS Adapter EJB on the remote application server. See one of the following:
   - *Deploying GIS Adapter EJB on a WebSphere Application Server*
   - *Deploying GIS Adapter EJB on a WebLogic Application Server*
   - *Deploying GIS Adapter EJB on a JBoss Application Server*
5. Test the client program on the remote application server with the business process.

## Configuring the EJB Adapter Listener Service

To configure the EJB Adapter Listener service, you must specify field settings in Sterling Integrator. There are no fields that must be defined in the Graphical Process Modeler (GPM).

## Sterling Integrator Configuration

The following table describes the fields used to configure the EJB Adapter Listener 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. |

| Field | Description |
|---|---|
| Select a Group | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |
| Listen Port(listenPort) | Port to which the remote application server sends requests. Be certain to select an available port for this parameter. Required. |
| Listen Hostname(ListenHostname) | Name or IP address of the host to which the remote application server sends requests. Optional. |
| Document Storage Type(docStorageType) | Defines how the document will be stored in the system. Required. Valid values are:<br>• System Default<br>• Database<br>• File System<br><br>**Note:** For more information about document storage types, see *Managing Services and Adapters*. |

## Deploying GIS Adapter EJB on a WebSphere Application Server

If your legacy tools reside in a WebSphere application server, follow these steps to install the J2EE application GIS Adapter EJB and deploy it in that server:

1. Copy the *install_directory*/client/ejb/GISInvokerEJB.jar file from the Sterling Integrator installation onto the machine that hosts your WebSphere application server.
2. Start the WebSphere server and go to the Administrative Console page.
3. Select **Install New Application** and enter the absolute path of the GISInvokerEJB.jar file.
4. Complete the installation procedure using the following guidelines:
   - Enter GISAdapterEJB for the Application Name
   - Enter GISAdapterEJB for the JNDI Name.
   - Update the access control list with the proper user information and be certain to disable the parameter Deny All.
5. Save the successful installation to the Master Configuration.
6. Navigate to **Environment** > **Naming** > **Name Space binding** and select **New**.
7. Configure the WebSphere environment using the following guidelines:
   - Obtain the name of the host that Sterling Integrator is installed on
   - Obtain the listenPort value used in the EJB Adapter Listener service configuration. This is the Listen Port parameter you used when configuring the service.
   - Create a new Name Space binding using the Binding Type String for the Sterling Integrator host and port.

Enter host for the Name in Name Space and your Sterling Integrator host name for the String Value.

Enter port for the Name in Name Space and your service configuration listenPort value for the String Value.

8. Save the environment to the Master Configuration.

9. You can now create a client program on your Websphere server that can initiate a Sterling Integrator business process.

See *EJB Client Program Example* for a sample Context Holder Object and a list of methods you can use in a client program and the properties they correspond to in Sterling Integrator.

## Deploying GIS Adapter EJB on a WebLogic Application Server

**Note:** This section is not needed if you chose to integrate with WebLogic during the Sterling Integrator installation process.

If your legacy tools reside in a WebLogic application server, follow these steps to install the J2EE application GIS Adapter EJB and deploy it in that server:

1. Copy the *install_directory*/client/ejb/GISInvokerEJB.jar file from the Sterling Integrator installation onto the machine that hosts your WebLogic application server.

2. Start the WebLogic server and go to the Administrative Console page. The console page can be found at http://host:*weblogic_port*/console.

3. Select **Deployments** > **EJB** under mydomain from the left toolbar menu.

4. Select the directory into which you want to upload the EJB adapter.

5. Click **upload it through your browser**.

6. Select the absolute path of the GISInvokerEJB.jar file and upload the file. The file GISInvokerEJB.jar should now be listed in the directory list.

7. Click the **Select** link for GISInvokerEJB.jar.

8. Highlight your server and move it from Available Servers to Target Servers.

9. Select the servers onto which you want to deploy GIS Invoker EJB.

10. Complete the installation procedure using GISAdapterEJB for the Application Name. There should be a Completed message in the Status column and GISAdapterEJB should appear under EJB in the left toolbar menu.

11. You can now create a client program on your WebLogic server that can initiate a Sterling Integrator business process.

See *EJB Client Program Example* for a sample Context Holder Object and a list of methods you can use in a client program and the properties they correspond to in Sterling Integrator.

**Note:** You can bind the host and port to JNDI in the java program by the following method:

```
InitialContext env = (InitialContext) ctx.lookup("java:comp/env");
ctx.rebind("host","hostname");
ctx.rebind("port","listenerPort");
```

## Deploying GIS Adapter EJB on a JBoss Application Server

**Note:** This section is not needed if you chose to integrate with JBoss during the Sterling Integrator installation process.

If your legacy tools reside in a JBoss application server, follow these steps to install the J2EE application GIS Adapter EJB and deploy it in that server:

1. Copy the *install_directory*/client/ejb/GISInvokerEJB.jar file from the Sterling Integrator installation into the deploy directory on the machine that hosts your JBoss application server.

2. Start the JBoss server and go to the Administrative Console page. The console page can be found at http://host:*jboss_port*/web-console.

3. Select **J2EE Domains** > **Manager** > **JBoss (http://www.jboss.org/) - 3.2.5** > **GISInvokerEJB.jar** > **GISInvoker** from the JBoss Management Console menu. You can see the state of the GISInvoker Bean and confirm that the bean was successfully deployed to the JBoss application server.

4. You can now create a client program on your JBoss server that can initiate a Sterling Integrator business process.

   **Note:** You can safely redeploy the application if it is already deployed. To undeploy it, just remove the archive from the *deploy* directory. You do not need to restart the server in either case.

See *EJB Client Program Example* for a sample Context Holder Object and a list of methods you can use in a client program and the properties they correspond to in Sterling Integrator.

**Note:** You can bind the host and port to JNDI in the java program by the following method:

```
InitialContext env = (InitialContext) ctx.lookup("java:comp/env");
ctx.rebind("host","hostname");
ctx.rebind("port","listenerPort");
```

## EJB Client Program Example

Before you can create a client program that will initiate a Sterling Integrator business process you must:

- Configure the EJB Adapter Listener service in Sterling Integrator.
- Create and check in the business process you want to initiate.
- Install and configure the GIS Adapter EJB on your remote J2EE application server.

The following lists Sterling Integrator-specific methods you can use in the client program:

| Purpose | Method |
|---|---|
| //Execute business process | `ContextHolder rch = ejb.submitContext (ch,"login","password");` |
| //Obtain business process status | `int wf_status = ejb.getState (rch,"admin","password");` |
| //Obtain process data | `ContextHolder con= ejb.retrieveContext (rch,"login","password");` |

The following is an example of a program you could create on your application server that would work with Sterling Integrator.

```
{\rtf1\ansi\ansicpg1252\deff0\deflang1033{\fonttbl{\f0\
fswiss\fcharset0 Arial;}}
\viewkind4\uc1\pard\f0\fs20\par
import javax.ejb.*;\par
```

```
import javax.naming.*;\par
import java.rmi.*;\par
import java.util.*;\par
import java.io.*;\par
\par
public class TestEJBAdapter \{\par
\par
public static void main(String arg[]) \{\par
\par
if(arg[0] == null || arg[1] == null || arg[2] ==null || arg[3] == null)\par
\{\par
        System.out.println("Missing arguemnts : testwl BPName , JNDI ,
GIS Host ,
        GIS Port");\par
        System.exit(0);\par
\}\par
\par
\par
try\{\par
\par
FileInputStream jndicfg = new FileInputStream(arg[1]);\par
\par
Properties p = new Properties();\par
p.load(jndicfg);\par
\par
// Obtain the Initial Context from Application Server\par
InitialContext ctx = new InitialContext(p);\par
\par
// doing lookup for GIS EJB adapter\par
Object obj = ctx.lookup("com.sterlingcommerce.GISInvoker");\par
\par
// Create  EJB home\par
\par
GISInvokerHome home = (GISInvokerHome)javax.rmi.PortableRemoteObject.
narrow(obj,
    Class.forName("com.sterlingcommerce.woodstock.services.ejb.client.
GISInvokerHome"));\par
\par
GISInvoker ejb = home.create();\par
\par
// Creating Context Holder object to launch Work flow in EJB\par
\par
        ContextHolder ch = new ContextHolder();\par
        ch.setName(arg[0]);\par
        ch.setMaxBPWaitTime(30000);\par
        ch.setWorkFlowDataOnError(true);\par
        ch.setBody("My body".getBytes("UTF-8"));\par
        ch.setBPName(arg[0]);\par
         System.out.println(ch.toString());\par
\par
// Executing Business Process\par
 ContextHolder rch = ejb.submitContext(ch,"admin","password");\par
  System.out.println("Executed BP "+rch);\par
\par
// Obtaining the Businss Process status  \par
  \par
  ejb.getState(rch,"admin","password");\par
  \par
 // Obtain the Context \par
ContextHolder con=  ejb.retrieveContext(rch,"admin","password");\par
\par
\}catch(Exception e) \{\par
    e.printStackTrace();\par
  \}\par
\par
```

```
\}\par
\par
\par
}
```

# Chapter 68. Encoding Conversion Service

The following table provides an overview of the Encoding Conversion service:

| System name | EncodingConversionType |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Translation |
| Description | Converts a file from one character encoding to another as part of an Sterling Integrator business process. A *character encoding* is a representation of data in a particular character set. A character set is a list of characters (letters, numbers, and symbols) that are recognized by computer hardware and software. Examples of encoding sets are American Standard Code for Information Interchange (ASCII), eight-bit Unicode Transformation Format (UTF-8), and Simplified Chinese (GBK). The encoding you are converting from must have characters similar to the encoding you are converting to. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | No |
| Application requirements | The encoding you are converting from must have characters similar to the encoding you are converting to. |
| Invocation | Runs as part of a business process. |

## Implementing the Encoding Conversion Service

To implement the Encoding Conversion service, complete the following tasks:
1. Activate your license for the Encoding Conversion service.
2. Create an Encoding Conversion service configuration.
3. Configure the Encoding Conversion service.
4. Use the Encoding Conversion service in a business process.

   When creating a business process, determine whether to extract the converted file from Sterling Integrator to a specific file location. To extract the file, follow the Encoding Conversion service with a File System adapter configuration in your business process.

## Configuring the Encoding Conversion Service

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

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| input_encoding | Encoding of the input file. Required. |
| output_encoding | Encoding to which you want to convert the file. Required. |

# Chapter 69. End Transaction Service

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

| System name | End Transaction Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services |
| Description | Ends a multi-step transaction in a business process. |
| Business usage | Use this service to mark the end of a multi-step transaction in a business process. All of the steps that occur between the Begin Transaction service and End Transaction service are part of one transaction and are committed or rolled back at the same time. This service should be used with the Begin Transaction service. |
| Usage example | Consider a customer ordering items online and using a shopping cart. The merchant's business process: <br><br> 1. Gets a document containing shopping cart information (customer information, product details). <br><br> 2. Inserts the customer information to the customer table. <br><br> 3. Inserts product details to the invoice table (product_id, quantity purchased). <br><br> 4. Updates the quantity of the product from the inventory table as the products were sold. <br><br> If step 4 fails (update the quantity), but steps 1 and 2 were committed. The merchant's inventory table is now inaccurate, because the items were reported as sold, but the items were not deducted from the inventory table. The Begin Transaction service and End Transaction service make it possible to have a transaction where all the steps are committed successfully or all the steps are rolled back. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The End Transaction service should be used with the Begin Transaction service. |
| Application requirements | Nothing external to Sterling Integrator is required to use this service. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | No |
| Returned status values | Returned status values: <br> • 0 – Success <br> • 1 – Error |
| Restrictions | N/A |
| Testing considerations | N/A |

## How the End Transaction Service Works

Use the End Transaction service to end a multi-step transaction.

**Business Process Example**

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

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

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

```
<process name="TestTran">
  <sequence>
    <operation name="Begin">
      <participant name="BeginTransactionService"/>
      <output message="Xout">
        <assign to="START_TRANSACTION">TRUE</assign>
        <assign to="DISTRIBUTED">TRUE</assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="XML Encoder">
       <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="root_element">root</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="LightweightJDBCAdapterType">
      <participant name="LightweightJDBCAdapterQuery"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="schedHour">-1</assign>
        <assign to="result_name">result</assign>
        <assign to="schedDay">-2</assign>
        <assign to="sql">INSERT INTO CUSTOMER_LW (CUSTOMER_ID,
             SIR_NAME, FIRST_NAME, MIDDLE_INITIAL, LAST_NAME, ADDRESS, CITY,
              STATE, ZIP, PHONE_NUM, EMAIL, ORDERS_PLACED, EMPLOYER,
              WORK_PHONE, WORK_EMAIL, WORK_ADDRESS1, WORK_ADDRESS2,
              WORK_CITY, WORK_STATE, WORK_ZIP) VALUES
              (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
               </assign>
        <assign to="param1" from="Testplan/Test4/Customer_ID/text()">
</assign>
        <assign to="paramtype1">Integer</assign>
```

```
        <assign to="param2" from="Testplan/Test4/SirName/text()"></assign>
        <assign to="paramtype2">String</assign>
        <assign to="param3" from="Testplan/Test4/FirstName/text()"></assign>
        <assign to="paramtype3">String</assign>
        <assign to="param4" from="Testplan/Test4/MiddleInitial/text()">
</assign>
        <assign to="paramtype4">String</assign>
        <assign to="param5" from="Testplan/Test4/LastName/text()"></assign>
        <assign to="paramtype5">String</assign>
        <assign to="param6" from="Testplan/Test4/Address/text()"></assign>
        <assign to="paramtype6">String</assign>
        <assign to="param7" from="Testplan/Test4/City/text()"></assign>
        <assign to="paramtype7">String</assign>
        <assign to="param8" from="Testplan/Test4/State/text()"></assign>
        <assign to="paramtype8">String</assign>
        <assign to="param9" from="Testplan/Test4/Zip/text()"></assign>
        <assign to="paramtype9">String</assign>
        <assign to="param10" from="Testplan/Test4/Phone_Num/text()"></assign>
        <assign to="paramtype10">String</assign>
        <assign to="param11" from="Testplan/Test4/Email/text()"></assign>
        <assign to="paramtype11">String</assign>
        <assign to="param12" from="Testplan/Test4/Orders_Placed/text()">
</assign>
        <assign to="paramtype12">Integer</assign>
        <assign to="param13" from="Testplan/Test4/Employer/text()"></assign>
        <assign to="paramtype13">String</assign>
        <assign to="param14" from="Testplan/Test4/Work_Phone/text()">
</assign>
        <assign to="paramtype14">String</assign>
        <assign to="param15" from="Testplan/Test4/Work_Email/text()">
</assign>
        <assign to="paramtype15">String</assign>
        <assign to="param16" from="Testplan/Test4/Work_Address1/text()">
</assign>
        <assign to="paramtype16">String</assign>
        <assign to="param17" from="Testplan/Test4/Work_Address2/text()">
</assign>
        <assign to="paramtype17">String</assign>
        <assign to="param18" from="Testplan/Test4/Work_City/text()"></assign>
        <assign to="paramtype18">String</assign>
        <assign to="param19" from="Testplan/Test4/Work_State/text()">
</assign>
        <assign to="paramtype19">String</assign>
        <assign to="param20" from="Testplan/Test4/Work_Zipcode/text()">
</assign>
        <assign to="paramtype20">String</assign>
        <assign to="pool">mysqlPool</assign>
        <assign to="row_name">row</assign>
        <assign to="schedMinute">-1</assign>
        <assign to="query_type">ACTION</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="LightweightJDBCAdapterType">
      <participant name="LightweightJDBCAdapterQuery"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="schedHour">-1</assign>
        <assign to="result_name">result</assign>
        <assign to="schedDay">-2</assign>
        <assign to="sql">INSERT INTO INVOICE CUSTOMER_ID,PRODUCT_ID,QTY)
VALUES
                (?, ?, ?) </assign>
        <assign to="param1" from="Testplan/Test4/Customer_ID/text()">
</assign>
```

```
                      <assign to="paramtype1">Integer</assign>
                      <assign to="param2" from="Testplan/Test4/PRODUCT_ID/text()"></assign>
                      <assign to="paramtype2">String</assign>
                      <assign to="param3" from="Testplan/Test4/QUANTITY/text()"></assign>
                      <assign to="paramtype3">Integer</assign>
                  </output>
                  <input message="inmsg">
                      <assign to="." from="*"></assign>
                  </input>
              </operation>
              <operation name="LightweightJDBCAdapterType">
                  <participant name="LightweightJDBCAdapterQuery"/>
                  <output message="LightweightJDBCAdapterTypeInputMessage">
                      <assign to="schedHour">-1</assign>
                      <assign to="result_name">result</assign>
                      <assign to="schedDay">-2</assign>
                      <assign to="sql">INSERT INTO PRODUCT QTY VALUES (?) WHERE
PRODUCT_ID=?
                      </assign>
                      <assign to="param1" from="Testplan/Test4/QUANTITY/text()"></assign>
                      <assign to="paramtype2">Integer</assign>
                      <assign to="param3" from="Testplan/Test4/PRODUCT_ID/Text()"></assign>
                      <assign to="paramtype3">String</assign>
                  </output>
                  <input message="inmsg">
                      <assign to="." from="*"></assign>
                  </input>
              </operation>
              <operation>
                  <participant name="EndTransactionService"/>
                  <output message="Xout">
                      <assign to="END_TRANSACTION">TRUE</assign>
                  </output>
                  <input message="Xin">
                      <assign to="." from="*"></assign>
                  </input>
              </operation>
          </sequence>
      </process>
```

## Implementing the End Transaction Service

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

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

## Output from Business Process to Service

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

| Parameter | Description |
|---|---|
| End_Transaction | Ends the transaction. Valid value is true (default) or false. Required. |
| Rollback_Transaction | Sets the transaction to rollback. Valid value is true or false (default). Optional. Business process logic may be used to determine that a transaction should not be committed. By calling an instance of this service with this parameter set to "TRUE", the transaction can be rolled back. |

# Chapter 70. Execution Control Service

The Execution Control service allows for the business process to dynamically change various execution parameters, primarily for performance. The following table provides an overview of the Execution Control service:

| System name | ExecutionControlService |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Process Controls |
| Description | The Execution Control service allows for the business process to dynamically change various execution parameters, primarily for performance. |
| Business usage | Use this service to dynamically affect the execution of a business process. A business process can be:<br><br>• Assigned to a different queue<br><br>• Assigned a different priority within its queue<br><br>• Assigned an absolute deadline (in milliseconds, as provided by the Timestamp Utility service)<br><br>For example, a received document will be held for batch processing, but a functional acknowledgement for it should be expedited. You can set the spawned business process for the functional acknowledgement at a higher priority than the batch process. |
| Usage example | At run time, this service can dynamically move within a business process to a lower or higher queue than the one assigned at check in time. Moving to a lower queue can free up threads for the highest priority processes. Additionally, the Execution Control service allows configuring for fewer steps in the process to be completed before re-entering the queue for an available thread in the workflow engine. For example, you can give a business process with a service level agreement time window processing priority by configuring it to hold onto its thread for many steps. In addition, you can move it up in the priority queue at a certain step. Conversely, you can configure a lower priority process to do very few steps before returning to the workflow engine for an available queue, thereby freeing resources for higher priority processing. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | No |
| Application requirements | The business process using this service should have appropriate queue priorities assigned. |
| Initiates business processes? | This service does not initiate business processes. It works within a business process to adjust the execution of the steps in its business process, and/or change its priority queue. |
| Invocation | Does not invoke a business process |

| Business process context considerations | This service works within the workflow context to dynamically reassign: |
|---|---|
| | • Process Execution Priority |
| | • Execution Queue Name |
| | • Execution Cycle Size |
| | • Deadline |
| Returned status values | Returned status values: |
| | • Success – Execution was successful |
| | • Error – Unexpected parameter failure (for unparsable data) |
| Restrictions | None |
| Persistence level | None |

## Implementing the Execution Control Service

To implement the Execution Control service, complete the following tasks:

1. Create an Execution Control service configuration.
2. Configure the Execution Control service.
3. Use the Execution Control service in a business process.

## Configuring the Execution Control Service

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

| Field | Description |
|---|---|
| deadline | Deadline time by which the business process should be executed, in milliseconds as provided by the Timestamp Utility service. Optional. |
| executionCycleSize | Number of steps AE will cycle. Required. Value values are 1 - 2147483647. Default is 10. |
| mandatoryNode | Cluster only. The specific node on which the business process must run. Optional. Specify any valid node number. You will receive an error if the specified node is not available.<br>**Note:** The Sterling Integrator initializes a business process workflow on any available node before moving the business process to the node specified in mandatoryNode. |
| preferredNode | Cluster only. The node on which you would like the business process to run. Optional. Specify any valid node number. If the specified node is not available, the service will use another available node. |
| priority | Priority suggestion within the queue to place a step within a business process. Required. Valid values are 1 - 9. Default is 4. |
| queueName | The queue to be placed into. Optional. Valid values are 1 - 9. |

| Field | Description |
|-------|-------------|
| yield | Allow the business process to yield its thread to a higher priority process. After the yield, it is placed back in the queue. Required. Valid values: <br><br>• true <br>• false (default) <br><br>**Note:** In order for changes to other settings, such as `executionCycleSize`, to take effect immediately, `yield` must be set to true. |

The fields configured in the GPM are passed from the business process to the Execution Control service.

## Business Process Example

The following sample BPML demonstrates the syntax and usage of the Execution Control Service:

```
<process name="TestECS">
  <sequence name="simple">
△△△△<assign to="TestStartStatus">Success</assign>
    <operation name="ExecutionControlServiceExample">
      <participant name="ExecutionControlService"/>
      <output message="Xout">
        <assign to="yield">false</assign>
        <assign to="queueName">6</assign>
        <assign to="priority">1</assign>
        <assign to="executionCycleSize">1000</assign>
      </output>
      <input message="Xin"/>
    </operation>
△△△△△<assign to="TestEndStatus">Success</assign>
  </sequence>
</process>
```

# Chapter 71. Export Service

The following table provides an overview of the Export service:

| System name | ExportService |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | The Export service automates exporting resources from the application. You specify the resources to be exported and the various options in the Export service input XML file. Alternatively, you can specify a resource tag (which defines all the resources to be exported) in the input XML file. Resource tags can be created using the Admin Console. See *Using Proper Syntax* for syntax information. You can export resources to an XML file or to an installable bundle. If you want to export resources to be used on a new installation of your application, use an installable bundle. You can then load the installable bundle as part of the new system installation. See *Using Proper Syntax* for syntax information. The input XML file can also be used to specify the export type: standard or advanced.<br><br>• Standard export copies non-versioned resources and the default version of versioned resources.<br><br>• Advanced export copies non-versioned resources and enables you to choose to export just default versions or all versions of versioned resources.<br><br>See *Using Proper Syntax* for syntax information. You can specify regular expressions to filter and selectively export the desired resources. See *Regular Expressions* for information. When exporting trading partner identities, PGP profiles, service configurations, and communities, you must use a security context utility to store a passphrase in the database. The passphrase is then used to encrypt the identity information when it is exported to the output XML file. See *Using the Security Context Utility* for information. |
| Business usage | The Export service can be used with the scheduler to automate the process of exporting application resources. You define the resources to be exported in an XML file. |
| Usage example | The Export service is typically used in the following situations:<br><br>• In a virtual VAN environment, trading partners will synchronize their business resources in an automated fashion by using a combination of export and import services.<br><br>• Certain application resources may need to be exported and sent to SCI customer support to assist in troubleshooting problems. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported platforms for your application |
| Related services | Import service |

| Application requirements | None |
|---|---|
| Initiates business processes? | No |
| Invocation | Generally, this service is invoked by the Scheduler. |
| Business process context considerations | None |
| Returned status values | Success –Successfully exported Error –there are two basic errors:<br><br>• There is no Primary Document, this service operates on the primary document<br><br>• Set passphrase using SecurityContext program for security context |
| Restrictions | None |
| Persistence level | Full |
| Testing considerations | Use the Export option from the Deployment > Resource Manager menu to export a set of resources. Save the status report from that export. Next, use the Export service to export the same set of resources. Compare the status reports for the two exports to determine if the Export service is successfully exporting all of the resources you specified. |

## How the Export Service Works

The Export service works based on an XML input file that you create. The input file must contain the context and identity values from a security context (only if trading partner identities are included), whether or not to export information based on resource tags, the resource tags or names, the output type (XML or installable bundle), and the export type (standard or advanced). It can contain much more information, however. There are many parameters and expressions that can be used to select just the resources you want. See *Regular Expressions* and *Options* for more information.

## Example Business Scenario

Your company has a trading partner community called Suppliers. You want to update your company's trading partner profile and send the suppliers a new service configuration to use when sending you data. To accomplish this:

1. Create a resource tag (Supplier_Update) that will contain the trading partner profile and the service configuration.

   From the Deployment menu, select **Resource Manager** > **Resource Tags** > **New Resource Tag**. Select the resources to associate.

2. Create a security context for this export operation:

   `install_dir\bin>securityContext.sh set Suppliers dec01 leopard`

   See *Using the Security Context Utility* .

3. Create a well-formed XML file to be used as input for the Export service. To test that the XML file is well-formed, open the file with Internet Explorer. Internet Explorer will open an XML file only if it is well-formed:

   ```
   <?xml version="1.0"?>
   <ExportConfiguration>
   <OutputFormatType>XML
   Document</OutputFormatType>
   <ExportResourcesBasedOnTagName>Yes</ ExportResourcesBasedOnTagName>
   <Tags>
   ```

```
<Tag
Name="Supplier_Update" />
</Tags>
<ExportType>Standard</ExportType>
<Passphrase>
<SecurityContext>Suppliers</SecurityContext>
<SecurityIdentity>dec01</SecurityIdentity>
<ExportCertificate>Yes</ExportCertificate>
</Passphrase>
</ExportConfiguration>
```

4. Create an Export service configuration and a File System adapter configuration to use in your business process. In the File System adapter configuration, specify the extraction folder (where the File system adapter will place the export file at the end of the business process).

5. Create a business process to export the files. The Export service and File System adapter will be used to export the resources and then extract the export file to a location on your system:
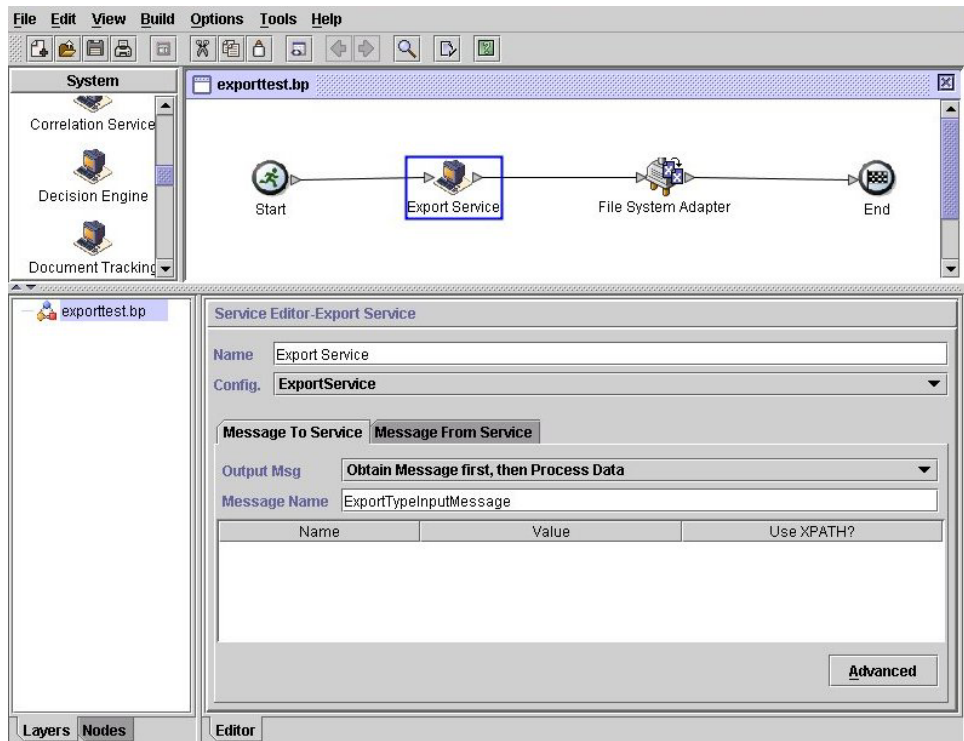


To enter the information for the Security context, click **Advanced** and add two entries, Context and Identity. Enter the values for each from the security context you created earlier:

Use the File System adapter configuration you created earlier in this business process. Select Extraction as the action in the GPM:
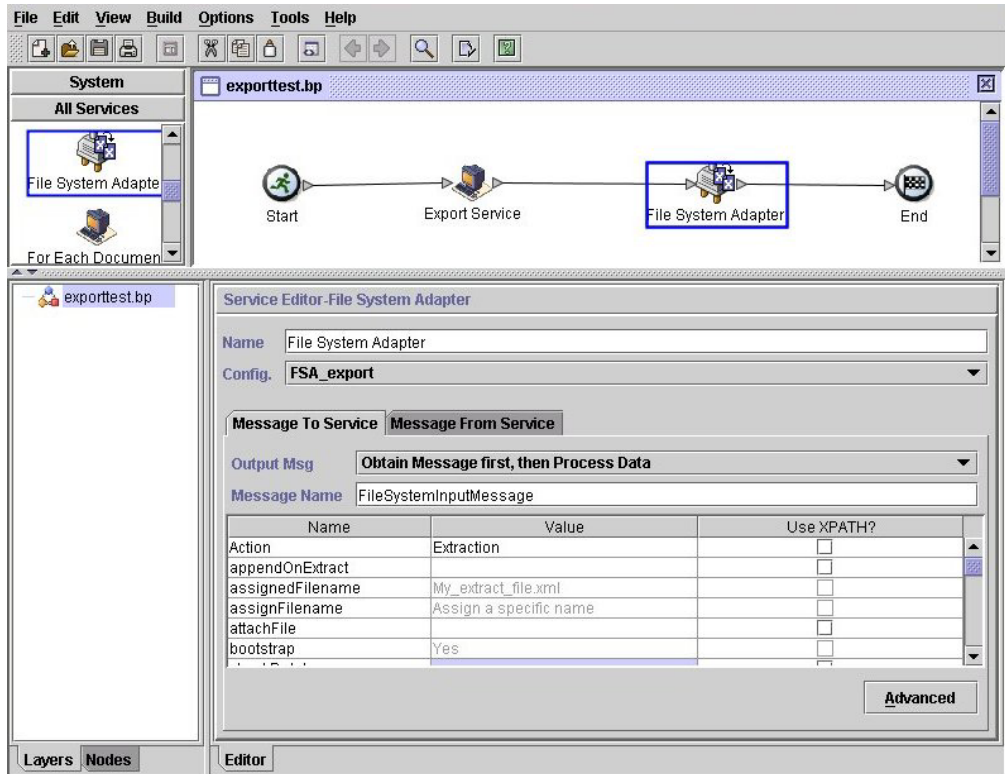
## Implementing the Export Service

To implement the Export service, complete the following tasks:

1. Identify the resources to be exported. If there are several types of resources (maps, services, trading partner information, for example), you can create a resource tag that includes all of the resources.

   For a list of resources that can be exported, see *Resources That Can Be Exported* .

2. Create a security context and passphrase for this export operation, if required for the type of data you are exporting. For information, see *Using the Security Context Utility* .

3. Create an XML file to use as input for the Export service. This file defines the resources and type of export. For information, see *Creating an Input XML File*.

4. Create an Export service configuration. See *Managing Services and Adapters*.

5. If you want to have the export file placed in a folder on your system, create a File System adapter configuration. Specify the folder to which the file should be extracted. For XML output, specify .xml as the file extension.

6. Use the Export service and File System adapter in a business process.

## Example Input XML Files

The following examples show how an XML file can be used as input for the Export service. The first example illustrates using resource tags; the second example illustrates using resource names.

## Example 1–Using Resource Tags

An easy way to create an export file is to create resource tags in the Admin Console and write an input file that refers to them. The file can include one or more resource tags. This example exports all the resources identified by two tags:

```xml
<?xml version="1.0"?>
<ExportConfiguration>
<OutputFormatType>XML Document</OutputFormatType>
<ExportResourcesBasedOnTagName>Yes</ExportResourcesBasedOnTagName>
<Tags>
    <Tag Name="bp1"/>
    <Tag Name="maps1"/>
</Tags>
<ExportType>Standard</ExportType>
<Passphrase>
<SecurityContext>secret1</SecurityContext>
<SecurityIdentity>secret2</SecurityIdentity>
<ExportCertificate>Yes</ExportCertificate>
</Passphrase>
</ExportConfiguration>
```

## Example 2–Using Resource Names

The following example specifies that resource names are used, rather than resource tags, and uses inclusions and exclusions to customize the data selected for the export:

```xml
<?xml version="1.0" ?>
 <ExportConfiguration>
   <OutputFormatType>XML Document</OutputFormatType>
   <ExportResourcesBasedOnTagName>No</ExportResourcesBasedOnTagName>
   <ExportType>Standard</ExportType>
   <Resources>
     <Resource>
       <Name>Business Process</Name>
       <PatternSet>
         <include>*</include>
         <exclude>AdvanceShipment</exclude>
       </PatternSet>
     </Resource>
    <Resource>
      <Name>Trading Partner Data</Name>
      <SubResources>
        <SubResource>
          <Name>Contracts</Name>
          <PatternSet>
            <include>pipContract3A12_initiator</include>
          </PatternSet>
        </SubResource>
      <SubResource>
        <Name>Identities</Name>
        <PatternSet>
          <include>111111111</include>
          <include>999999999</include>
        </PatternSet>
      </SubResource>
     </SubResources>
     </Resource>
   </Resources>
   <Passphrase>
      <SecurityContext>CompanyA</SecurityContext>
      <SecurityIdentity>Apr15</SecurityIdentity>
      <ExportCertificate>Yes</ExportCertificate>
   </Passphrase>
</ExportConfiguration>
```

## Using the Security Context Utility

The following types of data require a security passphrase when exported:
* Trading Partner Identities
* Trading Partner Envelopes
* PGP profiles
* Service configurations
* Communities
* Certain SSH resources: Host Identity Keys, User Identity Keys, and SSH Remote Profiles
* Web services
* Security tokens
* Proxy netmaps
* System digital certificates

The Export service works with the Security Context utility in the application. The utility is called securityContext.sh (for Unix) or securityContext.cmd (for Windows). It is located in the bin directory of your application installation. The security context utility can be used to store the passphrase and to retrieve it. The security context and security identity values are used to retrieve the associated passphrase. The passphrase is used to encrypt sensitive information (such as trading partner identities) in the export file.

The following information is used in the examples in this section:
* Context = CompanyA
* Identity = Apr15
* Passphrase = secret_phrase

When the utility is used to store the passphrase, provide the following three parameters:

```
securityContext.sh set CompanyA Apr15 secret_phrase
```

To check on the existence of a passphrase, use the following command:

```
securityContext.sh get CompanyA Apr15
```

The retrieve the list of contexts, use the following command:

```
securityContext.sh list_context
```

The context and identity need to be specified in the input XML file to direct the Export service to retrieve the appropriate passphrase from the database, as shown in the following example:

```
<Passphrase>
  <SecurityContext>CompanyA</SecurityContext>
  <SecurityIdentity>Apr15</SecurityIdentity>
  <ExportCertificate>Yes</ExportCertificate>
</Passphrase>
```

## Creating an Input XML File

In the XML files you create for input to the Export service, you can customize export operations by using the expressions and various options described in this section.

## Using Proper Syntax

This section illustrates the correct syntax to use when specifying resource tags, output types and standard or advanced exports.

Use the following syntax when referencing a resource tag:

```
<Tags><Tag Name="Test2" /></Tags>
```

Use the following syntax when specifying the output type for the export:
- For Install Bundle: <OutputFormatType>Install Bundle</OutputFormatType>
- For XML Document: <OutputFormatType>XML Document</OutputFormatType>

Use the following syntax when specifying the export type:
- For Standard : <ExportType>Standard</ExportType>

  The following example shows a sample standard export file:

```
<ExportType>Standard</ExportType>
<Resources>
<Resource>
<Name>Maps</Name>
  <PatternSet include= "^acme"/>
</Resource>
</Resources>
```

- For Advanced : <ExportType>Advanced</ExportType>

  The following example shows an advanced export that specifies all versions of versioned resources for map names that include acme should be exported:

```
<ExportType>Advanced</ExportType>
<Resources>
<Resource>
<Name>Maps</Name>
<PatternSet include= "^acme"/>
<AllVersions>true</AllVersions>
</Resource>
</Resources>
```

## Regular Expressions

You can use regular expressions to filter and selectively export resources such as business processes, contracts, and identities.

The following example XML fragment could be used in an XML file to export all business processes whose names begin with wssd:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet include= "^wssd"></PatternSet>
</Resource>
```

If using multiple includes, they are handled on separate lines, as shown in this example:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet>
       <include>^[a-m]</include>
       <include>wssd</include>
    </PatternSet>
</Resource>
```

The following example exports all business processes whose names begin with wssd except those whose names end with enu:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet include= "^wssd" exclude= "enu$"></PatternSet>
</Resource>
```

## Options

You can customize exports using the options described in this section:

### Characters

The following table contains the characters that you can use with the Export service:

| Character | Description |
|-----------|-------------|
| UnicodeChar | Matches any identical unicode character |
| \ | Used to quote a meta-character (like '*') |
| \\ | Matches a single '\' character |

### Character Classes

The following table contains the character classes that you can use with the Export service:

| Class | Description |
|-------|-------------|
| [abc] | Simple character class |
| [a-zA-Z] | Character class with ranges |
| [^abc] | Negated character class |

The following example exports all business processes whose names begin with letters A through M:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet include= "^[a-m]"></PatternSet>
</Resource>
```

The following example exports all business processes whose names begin with letters A through C:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet include= "^[abc]"></PatternSet>
</Resource>
```

### Portable Operating System Interface Character Classes

The following table contains the standard Portable Operating System Interface (POSIX) character classes that you can use with the Export service:

| Class | Description |
|-------|-------------|
| [:alnum:] | Alphanumeric characters. |
| [:alpha:] | Alphabetic characters. |

| Class | Description |
|---|---|
| [:blank:] | Space and tab characters. |
| [:cntrl:] | Control characters. |
| [:digit:] | Numeric characters. |
| [:graph:] | Characters that are printable and are also visible. (A space is printable, but not visible, while an `a' is both.) |
| [:lower:] | Lower-case alphabetic characters. |
| [:print:] | Printable characters (characters that are not control characters.) |
| [:punct:] | Punctuation characters (characters that are not letter, digits, control characters, or space characters). |
| [:space:] | Space characters (such as space, tab, and formfeed, to name a few). |
| [:upper:] | Upper-case alphabetic characters. |

The following example exports all business processes whose names contain a numeric character:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet include= "[:digit:]"></PatternSet>
</Resource>
```

The following example exports all business processes whose names begin with a lower-case character:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet include= "^[:lower:]"></PatternSet>
</Resource>
```

**Logical Operators**

The following table contains the logical operators that you can use with the Export service:

| Operator | Description |
|---|---|
| AB | Matches A followed by B |
| A|B | Matches either A or B |
| (A) | Used for subexpression grouping |
| (?:A) | Used for subexpression clustering (similar to grouping but no backrefs) |

The following example exports all business processes whose names begin with letters E, M, or W:

```
<Resource>
    <Name>Business Process</Name>
    <PatternSet>
    <include>^e|^m|^w</include>
    </PatternSet>
</Resource>
```

If you are searching for non-ASCII characters, such as é or ìå, save the input file as UTF-8.

## Resources That Can Be Exported

The following list contains the types of resources that can be exported using the Export service:

- Application configurations
- Business process
- Communities
- Contracts
- Control numbers
- Code lists
- Documents
- ebXML BPSS
- ebXML CPA
- Group permissions
- Groups
- HTML files
- Identities
- Images
- Javascript files
- JSP files
- Mailbox META
- Mailbox routing
- Mailboxes
- Maps
- Other files
- Perimeter servers
- Permissions
- Property files
- Schedules
- Schemas
- SCI packages
- Service configurations
- Stylesheets
- XML files
- Xslts
- User groups
- User permissions
- Users
- Web services
- WSDL

# Chapter 72. File System Adapter

The following table provides a high-level overview of the File System adapter:

| System name | FileSystem |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | Collects (imports) files from the file system and extracts (exports) files to the file system. The collected file becomes the primary document in a business process for file collection. A primary document is input to the File System adapter for file extraction. |
| Business usage | To read files from disk or write files to disk. |
| Usage example | Commonly used at the beginning of a business process to bootstrap a workflow by reading one or more files from disk and starting a business process. Another common use is to write files to disk for some external purpose.<br>**Note:** The term bootstrap is used in the Graphical Process Modeler to indicate that the File System adapter is used to start a business process after file collection. |
| Preconfigured? | No |
| Requires third party files? | None |
| Platform availability | All supported platforms for Sterling Integrator |
| Related services | None |
| Application requirements | None |
| Initiates business processes? | Yes, if you define a business process to start when you configure the File System adapter. The business process starts once files are collected. |
| Invocation | Normally, only the extraction side of the File System adapter is used in a business process, because you can configure the collection side of the File System adapter to start ("bootstrap") a business process. However, you can include the File System adapter directly in a business process to perform an explicit collection of files. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• Success – File System collection or extraction was successful.<br>• Error – File System collection or extraction was unsuccessful. |
| Restrictions | You must have read access to files and directories for file collection and write access to directories for file extraction. |
| Persistence level | System Default (Full) |
| Testing considerations | The best (and easiest) way to test this adapter is to set up a business process that only performs a file system extraction and specify that business process as the initial workflow to start (bootstrap). |

## How the File System Adapter Works

Use the File System adapter to collect (import) files from a file system into a business process and extract (export) files from a business process to a file system. You can configure the File System adapter to start a business process after files are collected from the file system or include the File System adapter in a business process flow. In addition, you can schedule the File System adapter to run at specific time intervals.

You can create multiple File System adapter configurations, one for each of several collection folders. Alternatively, you can use a single File System adapter configuration to point to different directories by specifying the directories for file collection and extraction explicitly in a business process. See *Usage Examples*.

The following sections describe a business scenario in which you could use the File System adapter, along with some sample solutions.

## Business Scenario

Your company receives a purchase order from a trading partner in EDI file format and the file is stored on the internal file system. You need to translate the EDI file into XML format and write the translated file to a local directory.

## Business Solution Example

The following approaches are used to solve the above business scenario.
1. Configure a File System adapter instance to be included in a business process to perform a file extraction.
2. Create a business process that translates the EDI file to XML format and then uses the File System adapter instance configured above to extract the resultant XML data to the file system
3. Configure a separate File System adapter instance to start the business process created in the previous step after an EDI file is detected in the collection folder. This File System adapter instance is also scheduled to run at 30-minute intervals.

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

## Graphical Process Modeler (GPM) Example

The following example shows a simple solution to the above business scenario using the GPM.
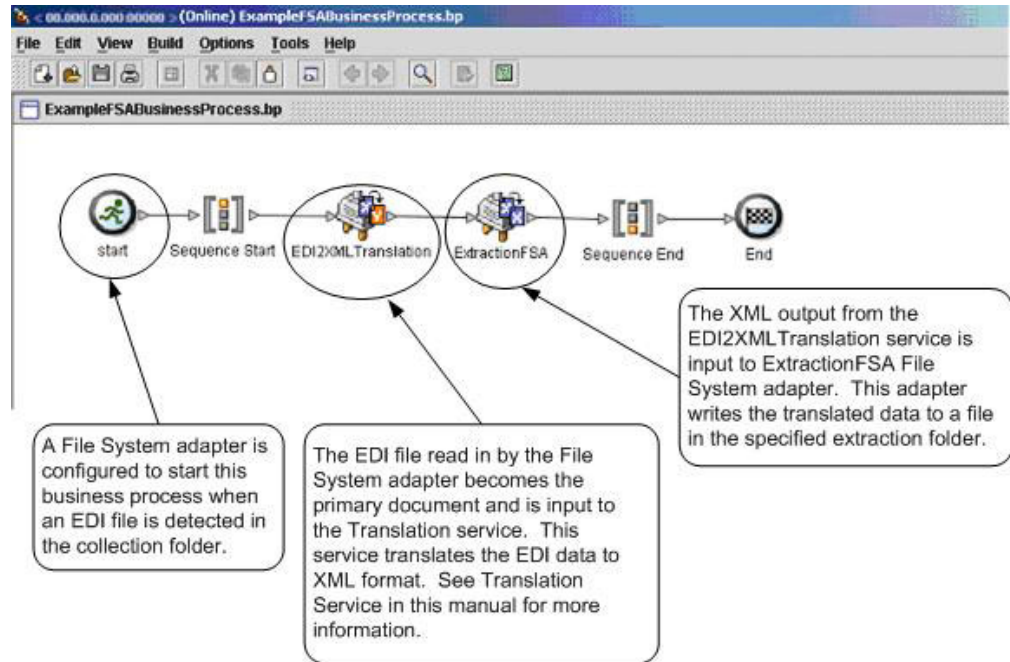
## Business Process Modeling Language (BPML) Example

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

```
<process name="ExampleFSABusinessProcess">
        <sequence name="Sequence Start">
        <operation name="EDI2XMLTranslation">
                <participant name="Translation"/>
                <output message="TranslationTypeInputMessage">
                        <assign to="exhaust_input">YES</assign>
                        <assign to="map_name">ExampleP0850</assign>
                        <assign to="output_to_process_data">NO</assign>
                        <assign to="validate_input">NO</assign>
                        <assign to="validate_input_against_dtd">NO</assign>
                        <assign to="validate_output">NO</assign>
                        <assign to="." from="*"/>
                </output>
                <input message="inmsg">
                        <assign to="." from="*"/>
                </input>
        </operation>
        <operation name="ExtractionFSA">
                <participant name="ExampleExtractionFSA"/>
                <output message="FileSystemInputMessage">
                        <assign to="Action">FS_EXTRACT<"assign>
                        <assign to="." from="*"/>
                </output>
                <input message="inmsg">
                        <assign to="." from="*"/>
                </input>
        </operation>
        </sequence>
</process>
```

FS_EXTRACT indicates you are writing a file to the file system.

## Implementing the File System Adapter

You can implement the File System adapter in three ways:
* Collect files within a business process.

- Extract files to the file system from a business process.
- Collect files and then start a new business process.

The information in this section applies to all three implementations.

Before you begin to implement a File System adapter, you need to collect the following information:
- The name of the business process (if the adapter is to start a business process)
- The directory path from which files are collected
- The directory path to which files are extracted

## Process Overview

To implement the File System adapter, complete the following tasks:
1. Create a File System adapter configuration.
2. Configure the File System adapter.
3. Create a business process to run after the File System adapter collects files, or create and enable a business process that includes the File System adapter (collecting or extracting files).
4. Test the business process and the adapter.
5. Run the business process.

## Configuring the File System Adapter

To create a File System adapter configuration, specify field settings in Sterling Integrator and in the GPM.

## File System Configuration

The following table describes the fields used to configure the File System adapter 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 service configuration. Required. |
| Description | Meaningful description for the service configuration, for reference purposes. Required. |
| Select a group | Group to associate with the adapter. Valid values:<br>• None – No group is selected<br>• Create New Group – Allows the creation of a new group<br>• Select Group – Select from a list of available groups |

| Field | Description |
| --- | --- |
| Collection folder (collectionFolder) | The name of the folder or subfolder on the same computer where Sterling Integrator is installed and where it collects (or picks up) files as part of a business process. If the path for the folder is not included as part of the name, the folder is assumed to be in the Sterling Integrator working directory. Required. **Note:** <br><br>• The deleteAfterCollect parameter in the GPM defaults to Yes. If you do not change the default value to No, files that are collected are deleted from the Collection Folder. The File System adapter does not copy the files it collects for processing. See *Graphical Process Modeler Configuration* for information about the deleteAfterCollect parameter. <br><br>• The collectionFolder parameter is read-only in the GPM. However, you can override this parameter using BPML. |
| Filename filter (filter) | Collect only files that match a specified filter within the collection folder. Optional. Examples include: <br><br>• *.txt (collects only .txt files) <br><br>• *.dat (collects only .dat files) <br><br>• EDI.* (collects only files named EDI with any file extension) <br><br>• EDI.txt (collect only files named EDI with a file extension of .txt) <br><br>**Note:** If there are multiple files in the collection folder and you leave this field blank, one of the following occurs: <br><br>• If the adapter is configured to start a business process, it processes all files placed in the collection folder. <br><br>• If the adapter is within a business process, it collects only the first file in the collection folder. <br><br>**Note:** If you specify this option using the File System adapter configuration, you cannot override the value using the GPM filter parameter. However, you can override this parameter using BPML. |
| Collect files from subfolders within and including the collection folder? (useSubFolders) | Whether to scan for files in subfolders of the collection folder. Required. Valid values: <br><br>• Yes – Collects files in the specified folder and all subfolders. <br><br>• No – Collects files in the specified folder only. <br><br>**Note:** This parameter is read-only in the GPM. |

| Field | Description |
|---|---|
| Use the absolute file path name for the document name? (keepPath) | Whether to keep the absolute path name of the files collected when assigning the document name. Required. Valid values:<br><br>• Yes – The absolute file path name is kept with the document in the business process. Choose this value if your business process requires the path information to precede the file name.<br><br>• No – Only the file name is kept with the document in the business process.<br><br>**Note:** An absolute path is a path that points to the same location regardless of the working directory or combined paths. It is usually written in reference to a root directory. For example, `c:\dir1\subdir1\somefile.txt` (Windows) and `/home/dir1/subdir1/somefile.txt` (UNIX) are examples of absolute paths to the file `somefile.txt`.<br>**Note:** This parameter is read-only in the GPM. |
| Start a business process once files are collected? (bootstrap) | Whether to start a business process using the File System adapter after files are collected. Required. Valid values:<br><br>• Yes – Starts the business process specified from the business process drop-down list.<br><br>**Note:** An instance of the business process is started for every file that matches the filtering criteria specified for file collection until the number of threads specified on the maxThreads parameter is reached. See *Graphical Process Modeler Configuration* for information about the maxThreads parameter.<br><br>• No – No business process will be started.<br><br>**Note:** This parameter is read-only in the GPM. |
| Business Process (initialWorkFlowId) | The business process to start after files are collected. Required when **Start a business process** is set to Yes. Valid values:<br><br>• Name of the business process to start<br><br>• Not Applicable<br><br>**Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify a business process using the configuration, you cannot override this value using the GPM initialWorkFlowId option. If you select Not Applicable, a business process can be selected in the GPM. In either case, you can override this parameter using BPML. |

| Field | Description |
|---|---|
| Document storage type (docStorageType) | Defines how the document will be stored in the system. Required. Valid values:<br>• System Default<br>• Database<br>• File System<br><br>**Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM DocStorageType option. However, you can override this parameter using BPML.<br>**Note:** For more information about document storage types, see *Selecting a Document Storage Method for Bootstrap Adapters*. |
| Obscure File Contents? (obscure) | Specifies whether to obscure the file contents when collecting. Does not work with "attachFile" or "importFile".<br>• Yes – File contents will be obscured<br>• No – File contents will not be obscured<br><br>**Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM Obscure option. However, you can override this parameter using BPML. |
| User Parameter 1 (userParm1) | A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm1. For more information, see *Example of Using User Parameters in a Business Process*.<br>**Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm1 option. However, you can override this parameter using BPML. |
| User Parameter 2 (userParm2) | A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm2.<br>**Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm2 option. However, you can override this parameter using BPML. |
| User Parameter 3 (userParm3) | A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm3.<br>**Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm3 option. However, you can override this parameter using BPML. |

| Field | Description |
|---|---|
| User Parameter 4 (userParm4) | A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm4. **Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm4 option. However, you can override this parameter using BPML. |
| User Parameter 5 (userParm5) | A user parameter that is passed to the bootstrapped workflow and placed in process data as UserParm5. **Note:** This field only displays as an option if **Start a business process once files are collected** is set to Yes. Additionally, if you specify this parameter using the configuration, you cannot override this value using the GPM userParm5 option. However, you can override this parameter using BPML. |
| Run As User | Applies to the scheduling of the business process. The Run As User field only displays as an option if **Start a business process once files are collected** 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. **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 can enable the scheduled run. |
| Use 24 Hour Clock Display | If selected, the adapter will use the 24-hour clock instead of the default 12-hour clock. |

| Field | Description |
|---|---|
| Schedule | Information about scheduling the business process after the File System adapter collects files. The Schedule field only displays as an option if **Start a business process once files are collected** is set to Yes. Valid values:<br><br>• Do not use schedule<br><br>  If you select this field, the adapter does not start a business process and does not run on a schedule.<br><br>• Run based on timer<br><br>  Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup.<br><br>• Run daily<br><br>  Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup.<br><br>• Run based on days of the week<br><br>  Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.<br><br>• Run based on days of the month<br><br>  Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |
| Extraction folder (extractionFolder) | The name of the folder or subfolder on the same computer where Sterling Integrator is installed and where it extracts (or writes) data from the primary document as part of a business process. If you do not include the file path for the folder as part of the name, the folder is assumed to be the Sterling Integrator working directory. Required.<br>**Note:** This parameter is read-only in the GPM. |
| Unobscure File Contents? (unobscure) | Whether to unobscure the file contents when extracting. Does not work with "exportFile". Valid values:<br><br>• Yes – File contents will be unobscured<br><br>• No – File contents will not be unobscured<br><br>**Note:** This parameter is read-only in the GPM. |

| Field | Description |
|---|---|
| Filenaming convention (assignFilename) | Whether to override the document file name and use the assigned file name. Required. Valid values:<br><br>• Use the original file name as the extracted file name – Keeps the names of the files.<br><br>**Note:** If the primary document has no document name, the adapter will use a default filename in the form of *nodename_yyyyMMddHHmmssSSS.dat*.<br><br>• Assign a specific name – Gives you the option to navigate to a screen and specify a different filename for the file extracted to the file system.<br><br>**Note:** This parameter is read-only in the GPM. |
| Filename (assignedFilename) | File name you want to assign, including the file name extension. The Filename field only displays if the *Filenaming convention* is set to Assign a specific name. Required. This field cannot be left blank. You can use "%^" to assign a unique file name in the format *nodename_yyyyMMddHHmmssSSS*.<br><br>For example, specifying %^.dat as the Filename assigns the name *nodename_20040203114020982.dat* to the file.<br>**Note:** This field can also be assigned in the GPM. If you select a filename using the File System adapter configuration, you cannot override it using the GPM assignedFilename parameter. However, you can override it using BPML. |
| Enable Service for Business Processes | Whether to enable the service for use by business processes. If not selected, the service will be disabled. For more information, see *Managing Services and Adapters.* |

## Example of Using User Parameters in a Business Process

The user parameters User Parameter 1 (UserParm1) through User Parameter 5 (UserParm5) in the File System adapter are places to store hard-coded values for use by other services. They are simple assign statements in the BPML.

For example, the Document Extraction service requires the following parameters when used for EDI XML extraction:

• XMLEDIEnvelopeStandard
• XMLRootTag
• XMLSenderIDPath
• XMLReceiverIDPath
• XMLAccepterLookupAliasPath

The File System adapter could be used in an intermediate business process to pass the parameters to the Document Extraction service through the use of User Parameters. The values for the above parameters would be stored in User Parameter 1 through User Parameter 5 in the File System adapter. The following assign statements would then be entered in the BPML code:

• UserParm1 = XMLEDIEnvelopeStandard
• UserParm2 = XMLRootTag
• UserParm3 = XMLSenderIDPath

- UserParm4 = XMLReceiverIDPath
- UserParm5 = XMLAccepterLookupAliasPath

The values stored in the User Parameters would be passed through to the Document Extraction service under the assigned parameter names.

## Graphical Process Modeler Configuration

The following screen shows the graphical view of the GPM parameters for the File System adapter. The dimmed values have been specified using the File System adapter configuration. The active fields are fields that cannot be configured in the Sterling Integrator or those that are being overridden. There are no fields to be configured on the **Message From Service** tab.

Screen 1 of 3

| Service Editor-File System Adapter | |
| --- | --- |
| **Name** | File System Adapter |
| **Config.** | **ExampleCollectionFSA** |

**Message To Service** | **Message From Service**

| **Output Msg** | **Obtain Message first, then Process Data** |
| --- | --- |
| **Message Name** | FileSystemInputMessage |

| Name | Value |
| --- | --- |
| Action | Collection |
| appendOnExtract | |
| assignedFilename | Output_Sample_Data_FufillOrder.xml |
| assignFilename | Use the original filename as the extracted filename |
| attachFile | |
| bootstrap | No |
| checkDelete | |
| collectionFolder | c:\collect |
| collectMultiple | |
| collectMultiplePDname | |
| collectMultiplePrefix | |
| collectZeroByteFiles | No |
| dbCollect | |

Screen 2 of 3

**Service Editor-File System Adapter**

| Name | File System Adapter |
|---|---|
| Config. | **ExampleCollectionFSA** |

| Message To Service | Message From Service |

| Output Msg | **Obtain Message first, then Process Data** |
|---|---|
| Message Name | FileSystemInputMessage |

| Name | Value |
|---|---|
| dbPurgeCollectMin | |
| deleteAfterCollect | No |
| docStorageType | System Default |
| extractionFolder | c:\extract |
| fileModTimeThreshold | 60 |
| filter | *.po |
| genReport | No |
| initialWorkFlowId | FIND_MESSAGE_CONSUMER |
| keepPath | Yes |
| maxCollect | |
| maxThreads | 10 |
| noFilesSetSuccess | No |
| obscure | No |

Screen 3 of 3

**Service Editor-File System Adapter**

| Name | File System Adapter |
|---|---|
| Config. | **ExampleCollectionFSA** |

| Message To Service | Message From Service |

| Output Msg | **Obtain Message first, then Process Data** |
|---|---|
| Message Name | FileSystemInputMessage |

| Name | Value |
|---|---|
| streamBufSize | |
| subCharsOnExtract | |
| unobscure | No |
| userParm1 | |
| userParm2 | |
| userParm3 | |
| userParm4 | |
| userParm5 | |
| useSubFolders | No |

The following example shows the corresponding BPML parameters for the File System adapter GPM parameters.

```
<process name="ExampleFileCollection">
 <operation name="File System Adapter">
  <participant name="ExampleCollectionFSA"/>
  <output message="FileSystemInputMessage">
```

```
            <assign to="." from="*"/>
            <assign to="Action">FS_COLLECT</assign>
            <assign to="collectZeroByteFiles">false</assign>
            <assign to="deleteAfterCollect">false</assign>
            <assign to="fileModTimeThreshold">60</assign>
            <assign to="filter">*.po</assign>
            <assign to="initialWorkFlowId">FIND_MESSAGE_CONSUMER</assign>
            <assign to="maxThreads">10</assign>
            <assign to="noFilesSetSuccess">false</assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"/>
          </input>
        </operation>
      </process>
```

The following table describes the fields used to configure the File System adapter in the GPM. This table contains only the fields that are configured in the GPM. The values in parentheses represent the corresponding BPML values. This information is provided for your reference.

| Field | Description |
|---|---|
| Config (participant name) | Name of the adapter configuration. Required. No default . |
| Action | Action that the File System adapter is to perform. Required. No default. Valid values:<br>• Collection (FS_COLLECT) – Files are collected or picked up from the specified folder.<br>• Extraction (FS_EXTRACT) – Files are extracted or written to the specified folder. |
| appendOnExtract | Whether to append the data if the extract file already exists. Normally, files are overwritten when extracting. This parameter allows you to append the data to the existing files instead. Valid values:<br>• Yes (true) – Data is appended to existing files.<br>• No (false) – Existing files are overwritten. Default |
| attachFile | Used to attach a file to a workflow as the primary document. The adapter does not perform any I/O and does not delete the file. Any valid filename is a valid value. |
| checkDelete | Determines if checking for deletion is possible before collecting files. Valid values:<br>• Yes (true) – Default<br>• No (false) |
| collectMultiple | Used to collect multiple files in non-bootstrap mode. Collected files are placed into process data.<br>• Yes (true)<br>• No (false) – Default |
| collectMultiplePDname | Used when collecting multiple files to specify which file will be the primary document. Any valid filename is a valid value. |

| Field | Description |
|---|---|
| collectMultiplePrefix | Specifies a prefix to be added to the document name. When multiple documents are created in process data, the documents are named Document1 through DocumentX. Multiple instances could overwrite the documents. You can use this prefix to differentiate the documents in different instances. For example, One instance could use the prefix Inst1_ and another instance could use the prefix Inst2_. The first instance would produce files named Inst1_Document*X* and the second instance would produce files named Inst2_Document*X*. The actual file name is placed as an attribute (filename=) in the document tag. The default value is FSA_. |
| collectZeroByteFiles (true/false) | Whether to collect zero-byte files. Valid values:<br>• Yes (true) – Zero-byte files are collected.<br>• No (false) – Zero-byte files are ignored. Default. |
| concatenateFiles | Used when the collectMultiple option is true and when the File System adapter is set in a non-bootstrap mode. The content of multiple non-zero byte files are concatenated into a single file, and is placed as the primary document. Optional. Valid values:<br>• Yes (true). Default.<br>• No (false) |
| dbCollect | If you set this field to true and the field deleteAfterCollect is set to true (which is the default), a database record will be written for every file collected. Before a file is collected, the database is checked to see if the file has already been collected. Optional. Valid values are Yes (true) and No (false). Default is No (false). |
| dbPurgeCollectMin | Used when dbCollect is set to Yes (true) to specify the number of minutes, from the time the database record is written, before the record is purged. Set this value slightly higher than the scheduled collection interval to prevent duplication before purging. Optional. Valid value is any valid (positive) integer value. Default is 1440 (one day). |
| deleteAfterCollect | Whether to delete the file after collection. Valid values:<br>• Yes (true) – File is deleted from the Collection folder after it is collected. Default.<br>• No (false) – File is left in the folder after it is read into Sterling Integrator. |
| fileModTimeThreshold | Sets the file modification time threshold (in seconds) for files to be collected. A file is collected only if the modification time of the file is older than the number of seconds specified. This prevents premature collection of a file. Defaults to 30 seconds if you do not specify a value. |

| Field | Description |
|---|---|
| filter | Collect only files that match a specified filter within the collection folder. Optional. Examples include:<br>• *.txt (collects only .txt files)<br>• *.dat (collects only .dat files)<br>• EDI.* (collects only files named EDI with any file extension)<br>• EDI.txt (collect only files named EDI with a file extension of .txt)<br><br>**Note:** If there are multiple files in the collection folder and you leave this field blank, one of the following occurs:<br>• If the adapter is configured to start a business process, it processes all files placed in the collection folder.<br>• If the adapter is within a business process, it collects only the first file in the collection folder.<br><br>**Note:** If you specified this option using the File System adapter configuration, this field will be read-only. However, you can override this parameter using BPML. |
| genReport | Determines whether a workflow status report is generated for all files regardless of whether they were successfully collected or not. Optional. Valid values:<br>• Yes (true) – Status report will be generated whether file collection was successful or unsuccessful. Default<br>• No (false) – Status report will only be generated if file collection is unsuccessful. |
| maxCollect | Sets the maximum number of files to collect. The default is -1 (unlimited). |
| maxThreads | Used for performance tuning to set the maximum number of threads used when collecting files. The default is ten threads. |
| noFilesSetSuccess | Used to determine the workflow status when no files are available to collect in non-bootstrap mode. Optional. Valid values:<br>• Yes (true) – The workflow status is set to Success even if no files exist in the specified collection folder during collection.<br>• No (false) – The workflow status is set to Error if no files exist in the specified collection folder during collection.<br><br>The default is No (false) |
| sortBy | When collectMultiple option is true and the File System Adapter is configured in a non-bootstrap mode, the files are sorted by File name or Modified date as selected in the GPM. The default setting places the files the way they were placed in the Collection folder. Optional. Valid Values: none. |
| streamBufSize | Used for performance tuning to override the default buffer size of 5k (5120). Optional. Valid value is any integer. |

| Field | Description |
|---|---|
| subCharsOnExtract | If the document name contains illegal filename characters, you can use this field to have them replaced with something else. An example would be if the document name was a GUID that contains colons ':' which are illegal in a Windows filename. In this case you would enter ":_" to replace all occurrences of the colon with an underscore. Optional. Must be entered in two character pairs with no delimiters or spaces. The first character is the one to be replaced, the second is the replacement itself. |

### Usage Examples

This section contains additional examples using the File System adapter for collecting and extracting files. Examples are included using both the GPM and BPML.

### File Collection

The following example using the GPM illustrates a business process that performs a file collection operation when the business process is started.



The following example illustrates the same business process using BPML.

```
<process name = "ExampleOverridenCollectionFolder">
 <operation name="File System Adapter">
  <participant name="ExampleCollectionFSA"/>
  <output message="FileSystemInputMessage">
   <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder">D:\Test\TestCollectionFolder</assign>
   <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
   <assign to="." from="*"></assign>
  </input>
 </operation>
</process>
```

FS_COLLECT
indicates file collection.

Use BPML to override the Collection Folder that was defined in the File System adapter configuration. This method lets you use the same file system adapter configuration to collect files from multiple directories.

## File Extraction

The following example using the GPM illustrates a business process that performs a file extraction operation when the business process is started. You could configure a File System adapter to start this business process after files are collected.



The following example illustrates the same business process using BPML.

```
<process name = "ExampleFileExtraction">
 <operation name="File System Adapter">
  <participant name="FS1"/>
  <output message="FileSystemInputMessage">
   <assign to="Action">FS_EXTRACT</assign>
   <assign to="assignedFilename">OverridenFileName</assign>
   <assign to="." from="*"></assign>
  </output>
  <input message="inmsg">
   <assign to="." from="*"></assign>
  </input>
 </operation>
</process>
```

FS_EXTRACT indicates file extraction.

Use BPML to override the assigned filename that was specified in the File System adapter configuration for the extracted file.

# Chapter 73. For Each Document Service

The For Each Document service incrementally processes a set of documents. Each time the service runs, it makes the next document in the document set the primary document. This enables business processes to contain loops that operate on each document in turn. The following table provides an overview of the For Each Document service:

| System name | For Each Document |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | The For Each Document service incrementally processes a set of documents. Each time runs, it makes the next document in the set the primary document. This enables business processes to contain loops that operate on each document in turn. |
| Business usage | To loop through multiple documents within a business process. |
| Usage example | Used when there are multiple documents within a business process that need to be processed. This service is used within the X12 Deenvelope and EDIFACT Deenvelope business processes to iterate document deenveloping over functional groups and transaction sets. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The Document Extraction service is often used in conjunction with the For Each Document service to extract individual documents out of a batch file. |
| Application requirements | None |
| Initiates business processes? | None |
| Invocation | Runs as part of a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• Success–The service has set the primary document appropriately.<br>• Error–The service encountered an unexpected condition that caused the iteration to fail. |
| Testing considerations | Create and test a business process that uses this service to incrementally process multiple documents. |

## How the For Each Document Service Works

Use the For Each Document service when you have multiple documents within the business process that all need to be processed.

The following sections describe a business scenario and sample solutions using the For Each Document service.

## Business Scenario

Your company has multiple records that are in a flat file format. You need to extract these records and process each one individually.

## Business Solution Example

The following approaches are used to solve the above business scenario.

1. Create maps required by the Document Extraction service to extract all the individual documents and place them in the business process.
2. Create a business process that:
   - Extracts the documents from the flat file using the Document Extraction service.
   - Uses the For Each Document service to iterate through all of the extracted documents .
   - Uses a Rule (more_split_docs) created through the Rule Manager and applied through the Edge Editor that checks if all documents have been processed.
   - Invokes a subprocess that processes each document.

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

**Note:** The focus in this example is on how to use the For Each Document service. The details on the subprocess are not included.

## GPM Example

The following example shows a solution to the above business scenario using the GPM:
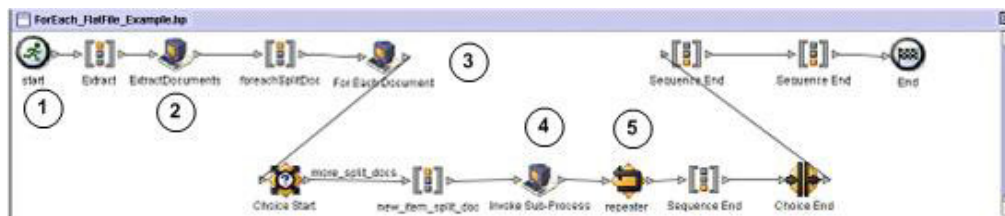


1. A flat file containing multiple records is passed to the business process as the primary document.
2. The Document Extraction service extracts all of the documents from the flat file and places them as individual documents in the process data of the business process.
3. The For Each Document service iterates through the documents, making each successive document the primary document.
4. A subprocess runs to process each of the documents.
5. The Rule (more_split_docs) is applied using the Edge Editor.

The following example shows the GPM configuration of the For Each Document service.
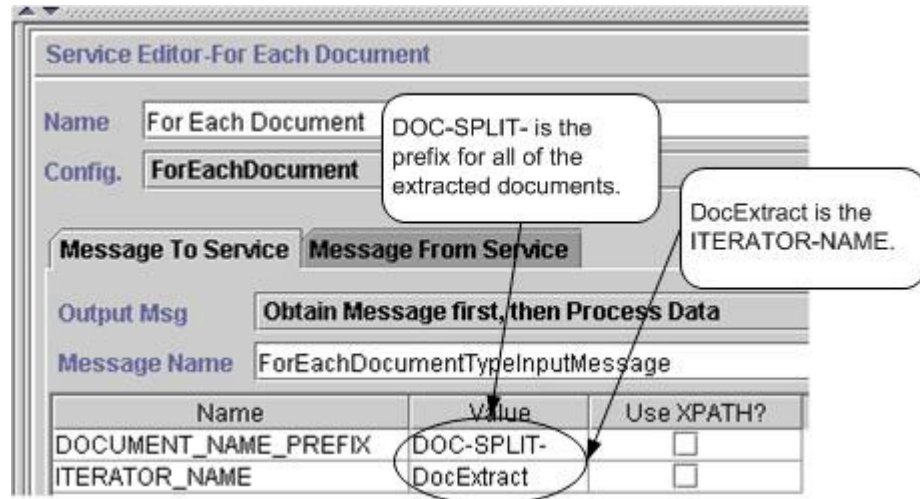
The following graphic shows the Rule Editor and Edge Editor configuration in the GPM for the business process.
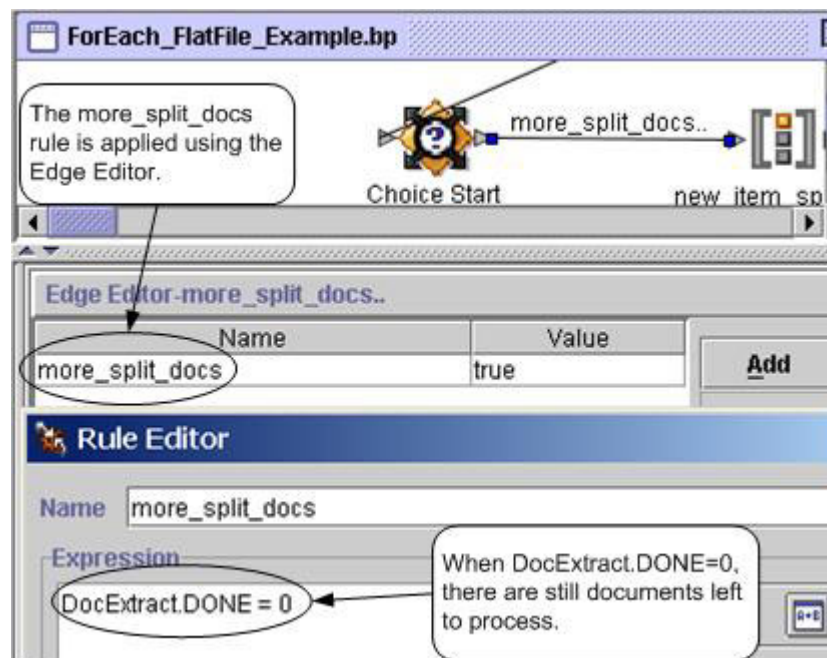


DocExtract.DONE is an output parameter from the For Each Document service.

The Rule (more_split_docs) is DocExtract.DONE = 0. This Rule is created using the Rule Manager and applied in the business process using the Edge Editor.

When this rule is true, processing continues. After all the documents are processed, DocExtract.DONE is set to 1 by the For Each Document service and processing stops.

## Business Process Modeling (BPML) Example

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

```
<process name="ForEach_FlatFile_Example">
  <rule name="more_split_docs">
    <condition>DocExtract.DONE = 0</condition>
  </rule>
  <sequence name="Extract">
    <operation name="ExtractDocuments">
      <participant name="DocumentExtractionService"/>
      <output message="DocumentExtractionTypeInputMessage">
        <assign to="BatchLikeDocuments">NO</assign>
        <assign to="DocExtractMapList">DocExtract_CTN DocExtract_ITM
</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
<sequence name="foreachSplitDoc">
      <operation name="For Each Document">
        <participant name="ForEachDocument"/>
        <output message="ForEachDocumentTypeInputMessage">
          <assign to="DOCUMENT_NAME_PREFIX">DOC-SPLIT-</assign>
          <assign to="ITERATOR_NAME">DocExtract</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
      <choice name="Choice Start">
        <select>
          <case ref="more_split_docs" activity="new_item_split_doc"/>
        </select>
<sequence name="new_item_split_doc">
          <operation name="Invoke Sub-Process">
            <participant name="InvokeSubProcessService"/>
            <output message="InvokeSubProcessServiceTypeInputMessage">
              <assign to="INVOKE_MODE">ASYNC</assign>
              <assign to="NOTIFY_PARENT_ON_ERROR">ALL</assign>
              <assign to="WFD_NAME"> ProcessNewItemRequest</assign>
              <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
              <assign to="." from="*"></assign>
            </input>
          </operation>
          <repeat name="repeater" ref="foreachSplitDoc"/>
        </sequence>
      </choice>
    </sequence>
  </sequence>
</process>
```

## Process Data

The following shows process data after the documents are extracted and the first document has been processed.

```
<ProcessData>
  <DOC-SPLIT-1 SCIObjectID="L2000-000248:f96c:fb591069a2:-60fb">
    <SenderID>03011</SenderID>
    <ReceiverID/>
    <AccepterLookupAlias/>
  </DOC-SPLIT-1>
  <DOC-SPLIT-2 SCIObjectID="L2000-000248:f96c:fb591069a2:-60fa">
    <SenderID>03012</SenderID>
    <ReceiverID/>
    <AccepterLookupAlias/>
  </DOC-SPLIT-2>
  <DOC-SPLIT-3 SCIObjectID="L2000-000248:f96c:fb591069a2:-60f9">
    <SenderID>03014</SenderID>
    <ReceiverID/>
    <AccepterLookupAlias/>
  </DOC-SPLIT-3>
  <DocExtract SCIObjectID="L2000-000248:f96c:fb591069a2:-60ea"/>
  <CONTRACT_FOUND>NO</CONTRACT_FOUND>
  <PrimaryDocument SCIObjectID="L2000-000248:f96c:fb591069a2:-60e9"/>
  <DocExtract.NAME>DOC-SPLIT-1</DocExtract.NAME>
  <DocExtract.INDEX SCIObjectID="L2000-000248:f96c:fb591069a2:-60e8"/>
  <DocExtract.DONE>0</DocExtract.DONE>
</ProcessData>
```

Three documents have been extracted:
DOC-SPLIT-1
DOC-SPLIT-2
DOC-SPLIT-3

The current primary document is DOC-SPLIT-1.

DocExtract.DONE=0 indicates there are more documents to process.

The following shows the process data after the business process has completed.

```
<ProcessData>
 .
 .
 .
  <CONTRACT_FOUND>NO</CONTRACT_FOUND>
  <DocExtract.DONE>1</DocExtract.DONE>
  <INVOKE_ID_LIST>46396</INVOKE_ID_LIST>
  <INVOKE_ID_LIST>46397</INVOKE_ID_LIST>
  <PrimaryDocument SCIObjectID="L2000-000248:f96c:fb591069a2:-60a1"/>
  <INVOKE_ID_LIST>46398</INVOKE_ID_LIST>
</ProcessData>
```

DocExtract.DONE=1 indicates all documents have been processed.

**Note:** When the Document Extraction service parameter PDToProcessData is set to No, process data will only contain the current DOC_SPLIT section. Each iteration through the For Each Document service will update process data with the current DOC_SPLIT and remove the previous split.

## Output from Service to Business Process

The following table describes the output from the For Each Document service to the business process. The parameters in the table, along with their values, are put into process data for use later in a business process.

**Note:** In the following table, I is replaced with the value of the ITERATOR_NAME parameter. If the ITERATOR_NAME were TEST, these parameters would be TEST, TEST.INDEX, TEST.NAME, and TEST.DONE.

| Parameter | Description |
|---|---|
| I | ArrayList containing the document data in the order that the documents are processed. |
| I.INDEX | The index of the document that was most recently made the primary document. |
| I.NAME | Name of the document that was most recently made the primary document. |
| I.DONE | Valid values:<br>• 0 – There are documents remaining to be processed.<br>• 1 – The iteration is complete. |

The following example shows the process data with the output parameters. The array list in this case is named DocExtract.

```
<ProcessData>
.
.
.
<DocExtract SCIObjectID="L2000-000248:f96c:fb591069a2:-60ea"/>
<DocExtract.NAME>DOC-SPLIT-1</DocExtract.NAME>
<DocExtract.INDEX SCIObjectID="L2000-000248:f96c:fb591069a2:-60e8"/>
<DocExtract.DONE>0</DocExtract.DONE>
.
.
.
</ProcessData>
```

## Usage Examples

The following business processes are predefined in Sterling Integrator:
- The X12Deenvelope business process
- The EDIFACTDeenvelope business process

## Implementing the For Each Document Service

To implement the For Each Document service, complete the following tasks:
1. If a configuration does not already exist, create a For Each Document service configuration. For information, see *Managing Services and Adapters*.

   **Note:** A For Each Document service is configured with the installation of Sterling Integrator.
2. Configure the For Each Document service. For information, see *Configuring the For Each Document Service*.
3. Use the For Each Document service in a business process.

## Configuring the For Each Document Service

To configure the For Each Document service, you must specify field settings in Sterling Integrator and in the GPM.

The following example shows the GPM parameters for the For Each Document service. There are no fields to be configured on the Message From Service tab.
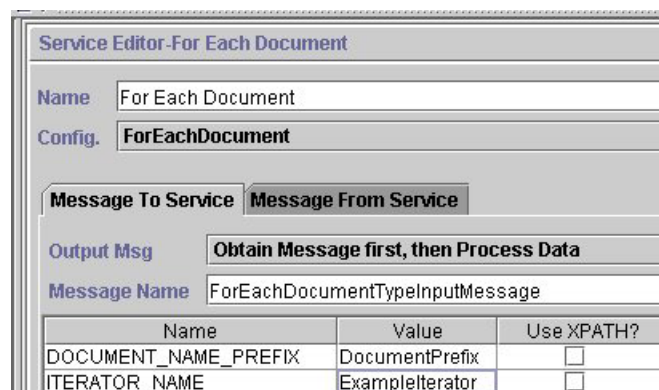
The following example shows the corresponding BPML parameters for the For Each Document service GPM parameters.

```
<operation name="For Each Document">
        <participant name="ForEachDocument"/>
        <output message="ForEachDocumentTypeInputMessage">
          <assign to="DOCUMENT_NAME_PREFIX">DocumentPrefix</assign>
          <assign to="ITERATOR_NAME">ExampleIterator</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
```

## GPM Configuration

The following table describes the fields used to configure the For Each Document service in the GPM:

| Field | Description |
|---|---|
| Config(participant name) | Name of the service configuration. |
| ITERATOR_NAME | Name used to identify this document iterator in the business process. This is the prefix for the tags that are generated by the service and is incremented as it loops through the documents. |
| DOCUMENT_NAME_PREFIX | Filter that restricts the documents included in the iteration. The filter is the prefix on each document name. When this parameter is used, only documents beginning with this prefix are processed. If DOCUMENT_KEY_PREFIX is used, this parameter is ignored. If neither this parameter or DOCUMENT_KEY_PREFIX is specified, all documents in the business process are processed by the For Each service. |
| DOCUMENT_KEY_PREFIX | Filter that restricts the documents included in the iteration. The filter is the prefix on the key in process data for each document. When this parameter is used, only documents that have keys beginning with this prefix are processed. The parameter DOCUMENT_NAME_PREFIX is ignored if this parameter is specified. If neither this parameter or DOCUMENT_NAME_PREFIX is specified, all documents in the business process are processed by the For Each service. |

# Chapter 74. FTP Client Adapter

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

| System name | FTP Client Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | Not available in the GPM |
| Description | This adapter is used in conjunction with the FTP Client services to send FTP requests to trading partners using a Perimeter server. The FTP Client adapter replaces the B2B FTP Client adapter, the FTP Send adapter, and the FTP GET adapter, which are being retired. The FTP Client adapter and its related services provide all the functionality of the former three adapters plus the following enhancements:<br>• Uses Perimeter Services<br>• Scriptable through BPML<br>• Highly scalable<br>• Supports large files (up to 15 GB)<br>• Designed to work easily with nearly any FTP server |
| Business usage | Use this adapter to send FTP requests to perform activities such as to **get** or **put** files into a directory on the trading partner's FTP server. |
| Usage example | A Sterling Integrator business process is executed that translates a document that must be sent to a trading partner. After the translation, Sterling Integrator looks up information about how to transport data to the trading partner in the trading profile. The trading profile specifies FTP as the transport protocol. Sterling Integrator then uses the FTP Client adapter to send the document to the trading partner. |
| Preconfigured? | No |
| Requires third party files? | Certicom SSL Library (currently available in Sterling Integrator) |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client GET service<br>• FTP Client LIST service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client QUOTE service<br>• FTP Client SITE service |

| | |
|---|---|
| Application requirements | An FTP 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.<br>**Note:** You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring. |
| Initiates business processes? | This adapter does not initiate business processes. |
| Invocation | This adapter is invoked from one of the FTP Client services used within a business process. |
| Business process context considerations | Business processes using FTP client services should not be marked as Auto Resume. These services require an established session which will no longer exist after a restart. |
| Returned status values | Returned status values:<br>• 0 – Success<br>• 1 – Error |
| Restrictions | None |
| Persistence level | None |
| Testing considerations | Test this adapter by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test. To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**.<br><br>2. Find FTP Server Adapter.<br><br>3. If not already selected, select the **Enabled** check box.<br><br>To test the FTP Client adapter, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Manager**.<br><br>2. Find the FTPClientDemoAllServices business process.<br><br>3. Run the FTPClientDemoAllServices business process with the following settings:<br>• Run As User = Admin<br>• Server filename = <install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml.<br><br>4. Verify that the business process runs successfullyDebug information for this adapter can be found in the FTP Client adapter and services log files. |

| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:<br><br>• FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document.<br><br>• FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |
|---|---|

## Business Scenario

You want to send a file to a trading partner, using FTP. The file must be placed in a specific directory on the trading partner's FTP server. After putting the file in the directory, you want to list the contents of the directory to verify that it was copied correctly.

## Business Solution Example

The following approaches are used to solve the above business scenario.

1. Create an FTP Client adapter configuration to be used by the FTP client services that will be included in a business process to send the file.

2. Create a business process that:
   • Opens an FTP communications session with the trading partner (using the FTP Client Begin Session service)
   • Changes to the correct directory on the trading partner's FTP server (using the FTP Client CD service)
   • Puts the file in the directory (using the FTP Client PUT service)
   • Generates a list of the directory contents to verify that the put command was successful (using the FTP Client LIST service)
   • Closes the FTP session (using the FTP Client End Session service)

This business solution is described for both the GPM and the BPML.

## GPM Example

The following example shows a solution to the above business scenario using the GPM:
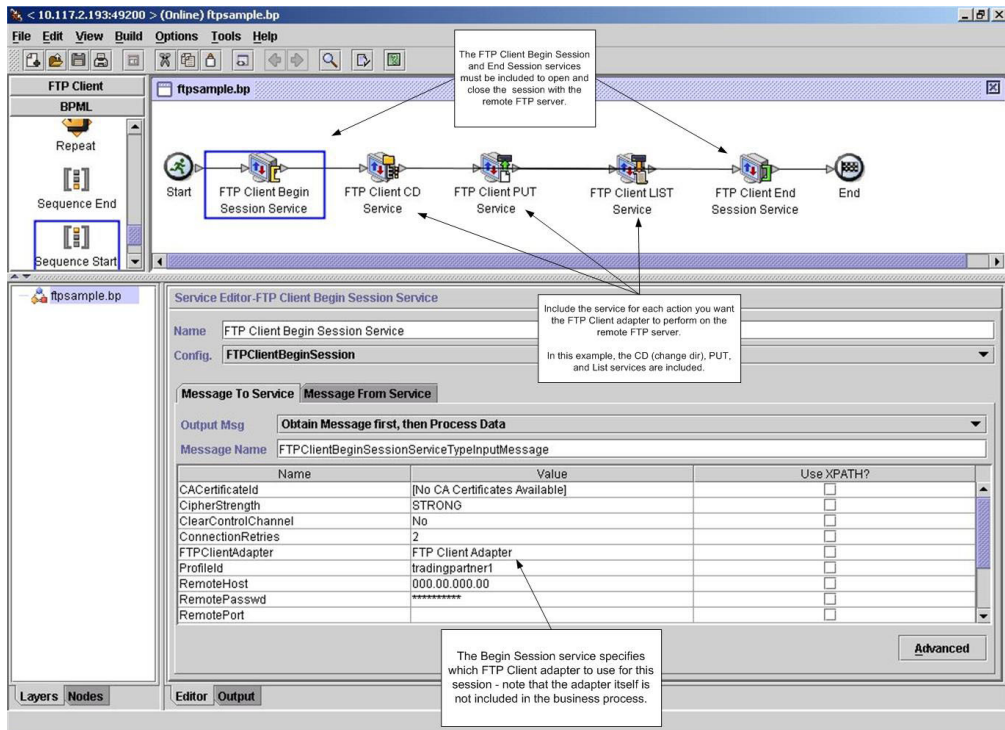
## Business Process Modeling Language (BPML) Example

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

```
<process name="default">
  <sequence>
    <operation name="FTP Client Begin Session Service">
      <participant name="FTPClientBeginSession"/>
      <output message="FTPClientBeginSessionServiceTypeInputMessage">
        <assign to="CipherStrength">STRONG</assign>
        <assign to="ClearControlChannel">NO</assign>
        <assign to="ConnectionRetries">2</assign>
        <assign to="FTPClientAdapter">FTPClientAdapter</assign>
        <assign to="ProfileId">tradingpartner1</assign>
        <assign to="RemoteHost">000.00.000.00</assign>
        <assign to="RemotePasswd">uuuuuuuu</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="FTP Client CD Service">
      <participant name="FTPClientCd"/>
      <output message="FTPClientCdServiceTypeInputMessage">
        <assign to="CdUp">YES</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
<operation name="FTP Client LIST Service">
      <participant name="FTPClientList"/>
      <output message="FTPClientListServiceTypeInputMessage">
        <assign to="ConnectionType">ACTIVE</assign>
```

```
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
      <operation name="FTP Client PUT Service">
        <participant name="FTPClientPut"/>
        <output message="FTPClientPutServiceTypeInputMessage">
          <assign to="DocumentId">doc12345</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
  <operation name="FTP Client End Session Service">
        <participant name="FTPClientEndSession"/>
        <output message="FTPClientEndSessionServiceTypeInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
</sequence>
</process>
```

## Implementing the FTP Client adapter

To implement the FTP Client adapter, complete the following tasks:

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

2. Configure the adapter. For information, see *Configuring the FTP Client adapter*.

## Configuring the FTP Client adapter

To configure the FTP Client adapter, 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. |

| Field | Description |
|---|---|
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br><br>• None – You do not want to include this configuration in a group at this time (default)<br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br><br>• 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.<br><br>For more information about service groups, see *Managing Services and Adapters*. |
| Perimeter Server | Select the Perimeter server to use with this adapter. Default is node1 & local. Required.<br>**Note:** You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring. |
| Min Threads | A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be less than or equal to Max Threads value. Valid value is any integer. Default is 3. Required. |
| Max Threads | A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be greater than or equal to Min Threads value. Valid value is any integer. Default is 6. Required. |
| Local Control Port Range | Any valid port numbers not being used by another application running on the system. A port in the specified range will be used to establish a control channel to the remote FTP server. Optional. |
| Local Data Port Range | Any valid port numbers that are not being used by another application running on the system. Contact your network administrator if the port numbers are restricted in any way. A port in the specified range will be used to establish a data connection to the remote FTP server. Optional. |

# Chapter 75. FTP Client Begin Session Service

The following table provides an overview of the FTP Client Begin Session service:

| System name | FTP Client Begin Session Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service is used to start an FTP session with an external trading partner to exchange business documents. |
| Business usage | Use this service to establish a session with a trading partner FTP server. |
| Usage example | A business process is executed that translates a document that must be sent to a trading partner. After the translation, Sterling Integrator uses the FTP Client Begin Session service to establish a session with the trading partner's FTP server. The Begin Session service works through a configuration of the FTP Client adapter. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All Sterling Integrator supported platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client LIST service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client SITE service<br>• FTP Client QUOTE service<br><br>To mask the values associated with the remote password parameter, use the Obscure Data - Process Data Values service in conjunction with the FTP Begin Session service. This service is presented in the GPM as Obscure Parameter on the All Services stencil. |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | The FTP Client Begin Session service allows you to specify a remote password. To obscure this password in process data for the business process, you must use the Obscure Data - Process Data Values service within the same business process. The Obscure Data - Process Data Values service masks the values associated with parameters. |

| Returned status values | Returned status values: |
| --- | --- |
| | • 0 – Success |
| | • 1 – Error |
| Restrictions | N/A |
| Persistence level | Default |
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test. To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**. |
| | 2. Find FTP Server Adapter. |
| | 3. If not already selected, select the **Enabled** check box. |
| | To test this service, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Manager**. |
| | 2. Find the FTPClientDemoAllServices business process. |
| | 3. Run the FTPClientDemoAllServices business process with the following settings: |
| | • Run As User = admin |
| | • Server filename = <install_directory>/installed_data/psftpclient/FTPClientDemoImport.xml. |
| | 4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule: |
| | • FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. |
| | • FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client Begin Session Service

To implement the FTP Client Begin Session service, complete the following tasks:

1. Create an FTP Client Begin Session service configuration. For information, see *Managing Services and Adapters*.
2. Use the FTP Client Begin Session service in a business process.

## Configuring the FTP Client Begin Session Service

You can set the following values in the trading partner profile and specify it in the ProfileId field, or you can set these values in an instance of the service to only apply for that instance. If specified in the FTP Client Begin Session service, the following values override those in the FTP trading partner profile:

- CACertificateId
- CipherStrength
- ConnectionRetries
- CharacterEncoding
- ConnectionTimeout
- RemoteHost
- RemotePasswd
- RemotePort
- RemoteUserId
- RetryDelay
- SSL
- SystemCertificateId

To configure the FTP Client Begin Session service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Name | Name this service will have in Sterling Integrator. Required. |
| Description | Description of service. Required. |
| Select a Group | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time. (default)<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |
| Config | Name of the service configuration. Select FTPClientBeginSession. |
| CACertificateId (trusted_root) | Select from the list of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID. Required if SSL = IMPLICIT or SSL = EXPLICIT. Check an SSL certificate into the application to make available from the list. |

| Field | Description |
|---|---|
| CipherStrength | The level of encryption to apply to the data that flows through the socket connection. Optional. Valid values are:<br>• ALL – WEAK or STRONG is accepted<br>• WEAK – 40 bit encryption is required<br>• STRONG – 40 bit or higher encryption is required (default) |
| ClearControlChannel | Indicates if information that travels across the control channel should be clear. Optional. Valid values are Yes and No. |
| ConnectionRetries | The number of times the service will try to connect to the trading partner system. Connection retries occur only with TCP/IP related issues. Optional. Valid value is any numeric value. While using the ConnectionRetries parameter, set the ResponseTimeout value to wait longer than the total time for RetryDelay and ConnectionRetires parameters. This setting allows the business process to remain active to perform the retries before the session times out and terminates. The following example illustrates the setting where the value of the ResponseTimeout (300) is greater than the total time taken by RetryDelay and ConnectionRetires parameters (30*5=150):<br><br>`<assign to="ResponseTimeout">300</assign>`<br>`<assign to="RetryDelay">30</assign>`<br>`<assign to="ConnectionRetries">5</assign>` |
| CharacterEncoding | The encoding format used to encode all outgoing commands and incoming data. If CharacterEncoding is not specified, the default system encoding will be used. Valid value is any valid encoding scheme supported by Java. Optional. |
| ResponseTimeout | Maximum number of seconds the FTP Client Begin Session service waits for the trading partner system to respond before the session times out and terminates. You can also set this parameter from a trading profile. The value you specify in the FTP Client Begin Session service overrides the value you specify in the trading partner profile. Optional. Valid value is any numeric value. Default value is 30 seconds. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client Begin Session service resets the value to 1 second. |
| FTPClientAdapter | Select the FTP Client adapter for this service to use when beginning sessions with an FTP server. Required. |
| ProfileId | Trading partner profile identification. Optional. Valid value is any valid profile ID. |
| UsingRevealedPasswd | Indicates whether the password sent to the service is unobscured. Valid value is True or False. Default is False. Optional. |
| RemoteAccount | FTP remote login account. Valid value is valid login account. There is no default value. Optional. |

| Field | Description |
|---|---|
| RemoteHost | External trading partner host system (FTP server IP address or DNS name). Required. Valid value is a valid IP Address or DNS name. |
| RemotePasswd | FTP remote login password. Optional. **Note:** For the password to be masked in process data, the Obscure Data - Process Data Values service must also be used in the same business process. The name used to store the password must be the same as the specified RemoteUserId. |
| RemotePort | External trading partner port number. Required. |
| RemoteUserId | FTP remote login user name. Optional. |
| RetryDelay | The delay (in seconds) the adapter will wait before retrying. Optional. Valid values are numeric values between 1 and 7200. The default value is 1 second. |
| SSL | Determines SSL socket negotiation. Optional. Valid values are:<br>• SSL_IMPLICIT – FTP server expects and requires SSL to happen automatically at the time of connection. CACertificateId is required.<br>• SSL_EXPLICIT – FTP client requests SSL and a secure connection is negotiated. CACertificateId is required.<br>• SSL_NONE – Connection will not use SSL. (default) |
| SystemCertificateId | Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. This certificate confirms the identity of the client to the server. Required if SSL = SSL_IMPLICIT or SSL_EXPLICIT and the server requires client authentication. Obtain the certificate from your trading partner. Check it into Sterling Integrator from the Admin menu selecting Trading Partner > Digital Certificates > System to make it available in this list. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br><br>Default is on. Optional. |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client Begin Session service to the business process:

| Parameter | Description |
|---|---|
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. |
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. |

# Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client Begin Session service:

| Parameter | Description |
|---|---|
| CACertificateId(trusted_root) | List of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID. |
| CipherStrength | The level of encryption to apply to the data that flows through the socket connection. Valid values are:<br>• ALL – WEAK or STRONG is accepted<br>• WEAK – 40 bit encryption is required<br>• STRONG – 40 bit or higher encryption is required |
| ClearControlChannel | Indicates if information that travels across the control channel should be clear. Valid values are Yes and No. |
| ConnectionRetries | The number of times the service will try to connect to the trading partner system. Valid value is any numeric value. 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):<br><br>`<assign to="ResponseTimeout">300</assign>`<br>`<assign to="RetryDelay">30</assign>`<br>`<assign to="ConnectionRetries">5</assign>` |
| ResponseTimeout | Maximum number of seconds the FTP Client Begin Session service waits for the trading partner system to respond before the session times out and terminates. You can also set this parameter from a trading profile. The value you specify in the FTP Client Begin Session service overrides the value you specify in the trading partner profile. Optional. Valid value is any numeric value. Default value is 30 seconds. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client Begin Session service resets the value to 1 second. |
| FTPClientAdapter | Select the FTP Client adapter for this service to use when beginning sessions with an FTP server. |
| ProfileId | Trading partner profile identification. Valid value is any valid profile ID. |
| RemoteHost | External trading partner host system (FTP server IP address or DNS name). Valid value is a valid IP Address or DNS name. |
| RemotePasswd | FTP remote login password. |
| RemotePort | External trading partner port number. |
| RemoteUserId | FTP remote login user name. |

| Parameter | Description |
|---|---|
| RetryDelay | The delay (in seconds) the adapter will wait before retrying. Valid values are numeric values between 1 and 7200. The default value is 1 second. |
| SSL | The SSL flag that determines SSL socket negotiation. Valid values are:<br>• SSL_IMPLICIT – FTP server expects and requires SSL to happen automatically at the time of connection.<br>• SSL_EXPLICIT – FTP client requests SSL and a secure connection is negotiated.<br>• SSL_NONE – Connection will not use SSL. |
| SystemCertificateId | Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. Valid value is any alphanumeric string. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• errononly – persists the transcript only when an error occurs<br>• on – always persists the transcript |

## Business Process Example

The following example business process illustrates using the FTP Client Begin Session service:

```
<process name="FtpExample">
 <sequence>
   <operation name="Obscure Password">
     <!-- insert obscured password into process data -->
     <participant name="FTPClientObscureParameter"/>
     <output message="outmsg">
     <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
     <assign to="." from="*"></assign>
     </input>
   </operation>
   <operation name="FTP Client Begin Session Service">
     <participant name="FTPClientBeginSession"/>
     <output message="FTPClientBeginSessionServiceTypeInputMessage">
       <assign to="FTPClientAdapter">FTPClientAdapter</assign>
       <assign to="RemoteHost">hostb</assign>
       <assign to="RemoteUserId">admin</assign>
       <!-- copy obscured password from process data to service -->
       <assign to="RemotePasswd" from="admin/text()"></assign>
       <assign to="RemotePort">30651</assign>
       <assign to="CipherStrength">STRONG</assign>
       <assign to="SSL">SSL_MUST</assign>
       <assign to="CACertificateId">FTP Server CA Cert</assign>
       <assign to="SystemCertificateId">FtpClientSystemCert</assign>
       <assign to="RemoteUserId">admin</assign>
       <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
       <assign to="FTPClientBeginSessionServiceResults" from="*"></assign>
     </input>
   </operation>
[[end session here]]
</process>
```

The following example shows how to use the revealObscured function when the user ID contains domain or special characters. First, create a name in the Obscure Service without any special characters and assign the appropriate password to it. In the following example, the name created in the Obscure Service is "abcd" while the userid is sgp-abcd\abcd.

The parameter to the revealObscured() function is the node containing the obscured password. The function uses the node name as the key and the node value as the obscured password when unobscuring.

```
<operation name="Obscure FTP client password">
      <participant name="FTPClientObscureParameter"/>
      <output message="outmsg">
 <assign to="." from="*"/>
      </output>
      <input message="inmsg">
 <assign to="ObscureResult" from="*"/>
      </input>
</operation>
<operation name="PS FTP BEGIN SESSION SERVICE">
      <participant name="FTPClientBeginSession"/>
      <output message="BeginSessionRequest">
           ............
         <assign to="RemoteUserId">sgp-abcd\abcd</assign>
         <assign to="UsingRevealedPasswd">true</assign>
         <assign to="RemotePasswd" from="revealObscured(
ObscureResult/abcd)"/>
           ..........
      </output>
      <input message="inmsg">
         <assign to="FtpBeginSessionServiceResults" from="*"/>
      </input>
</operation>
```

# Chapter 76. FTP Client CD Service

The following table provides an overview of the FTP Client CD service:

| System name | FTP Client CD Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service is used to change directories on the trading partner's FTP server. |
| Business usage | Use this service to change directories on the trading partner FTP server system. |
| Usage example | A 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. Sterling Integrator uses the FTP Client CD service, working through the FTP Client adapter, to change directories to the one where the document is located. Sterling Integrator can then use the FTP Client GET service to retrieve the document. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client LIST service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client SITE service<br>• FTP Client QUOTE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is initiated from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0 –Success<br>• 1 –Error |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.<br><br>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**.<br><br>2. Find FTP Server Adapter.<br><br>3. If not already selected, select the **Enabled** check box.<br><br>To test this service, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Manager**.<br><br>2. Find the FTPClientDemoAllServices business process.<br><br>3. Run the FTPClientDemoAllServices business process with the following settings:<br>• Run As User = Admin<br>• Server filename = <install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml.<br><br>4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:<br>• FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document.<br>• FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client CD Service

To implement the FTP Client CD service, complete the following tasks:

1. Create an FTP Client CD service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client CD service. For information, see *Configuring the FTP Client CD Service*.
3. Use the FTP Client CD service in a business process.

### Configuring the FTP Client CD Service

To configure the FTP Client CD service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| CdUp | • YES-Causes the FTP protocol command CDUP to be sent to the server, which results in the current directory moving one level up.<br>• NO-Causes the FTP protocol command CWD + Directory to be sent to the server. Default is NO. |
| Directory | The directory to change to. Can also be a relative directory depending on server capabilities. Required if CdUp is No.<br>**Note:** If CdUp=YES, this parameter is ignored. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client CD service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• errorronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br><br>Default is on. Optional. |

### Output from Service to Business Process

The following table contains the parameters passed from the FTP Client CD service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client CD service:

| Parameter | Description |
|---|---|
| CdUp | Valid values are:<br>• YES-Causes the FTP protocol command CDUP to be sent to the server, which results in the current directory moving one level up.<br>• NO-Causes the FTP protocol command CWD + Directory to be sent to the server. |
| Directory | The directory to change to. Can also be a relative directory depending on server capabilities. Required if CdUp is NO.<br>**Note:** If CdUp=YES, Directory will be ignored. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client CD service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Business Process Example

The following example business process illustrates using the FTP Client CD service:

```
 <sequence>
[[Insert FTP Client Begin Session]]
<operation name="FTP CD SERVICE">
<participant name="FTPClientCd"/>
<output message="CdRequest">
<assign to="SessionToken" from="/ProcessData/
FtpBeginSessionServiceResults/SessionToken/text()">
</assign>
<assign to="Directory">home/username/documents</assign>
</output>
<input message="inmsg">
<assign to="FTPClientCdResults" from="*"></assign>
```

```
     </input>
    </operation>
   [[ Insert FTP Client End Session ]]
</sequence>
```

# Chapter 77. FTP Client DELETE Service

The following table provides an overview of the FTP Client DELETE service:

| System name | FTP Client DELETE Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | Use to delete a document in a specified directory on the trading partner's FTP server. |
| Business usage | Use this service to delete a document on the trading partner system when FTP is used. |
| Usage example | A Sterling Integrator business process is executed that requires the removal of a document on the trading partner's system. Sterling Integrator uses the FTP Client DELETE service, working through the FTP Client adapter, to remove the specified document from the directory on the trading partner system. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client End Session service<br>• FTP Client LIST service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client SITE service<br>• FTP Client QUOTE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0–Success<br>• 1–Error |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.

To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console:

1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**.

2. Find FTP Server Adapter.

3. If not already selected, select the **Enabled** check box.

To test this service, perform the following steps from the Sterling Integrator Admin Console:

1. Choose **Business Processes** > **Manager**.

2. Find the FTPClientDemoAllServices business process.

3. Run the FTPClientDemoAllServices business process with the following settings:
- Run As User = Admin
- Server filename = <install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml.

4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:
- FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document.
- FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client DELETE Service

To implement the FTP Client DELETE service, complete the following tasks:

1. Create an FTP Client DELETE service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client DELETE service. For information, see *Configuring the FTP Client DELETE Service*.
3. Use the FTP Client DELETE service in a business process.

## Configuring the FTP Client DELETE Service

To configure the FTP Client DELETE service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| RemoteFileName | The name of the file to delete from the remote trading partner directory. Optional. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client DELETE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br><br>Default is on. Optional. |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client DELETE service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client DELETE service:

| Parameter | Description |
|---|---|
| RemoteFileName | The name of the file to delete from the remote trading partner directory. Optional. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client DELETE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• errononly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Business Process Example

The following example business process illustrates using the FTP Client DELETE service.

```
 <sequence>
  [[ Insert FTP Client Begin Session ]]
   <operation name="FTP DELETE SERVICE">
    <participant name="FTPClientDelete"/>
    <output message="DeleteRequest">
       <assign to="SessionToken"
from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
       </assign>
       <assign to="RemoteFileName">Filename.txt</assign>
    </output>
    <input message="inmsg">
       <assign to="FtpGetServiceResults" from="*"></assign>
    </input>
   </operation>
  [[ Insert FTP Client End Session ]]
 </sequence>
```

# Chapter 78. FTP Client End Session Service

The following table provides an overview of the FTP Client End Session service:

| System name | FTP Client End Session Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service is used to end an FTP session with an external trading partner FTP server. |
| Business usage | You would use this service as the last functional activity in a business process that sends an FTP request to a trading partner. This service can only be used if the FTP Client Begin Session service was used previously in the business process. |
| Usage example | A Sterling Integrator business process is executed that translates a document that must be sent to a trading partner. After the translation, Sterling Integrator begins a session with the trading partner using the FTP Client adapter, sends the document, then ends the session using the FTP Client End Session service. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client LIST service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client SITE service<br>• FTP Client QUOTE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service can be invoked in a business process to terminate a session with an FTP server. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0 –Success<br>• 1 –Error |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test. |
| | To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**. |
| | 2. Find FTP Server Adapter. |
| | 3. If not already selected, select the **Enabled** check box. |
| | To test this service, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Manager**. |
| | 2. Find the FTPClientDemoAllServices business process. |
| | 3. Run the FTPClientDemoAllServices business process with the following settings: |
| | • Run As User = Admin |
| | • Server filename = <install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml. |
| | 4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule: |
| | • FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. |
| | • FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client End Session Service

To implement the FTP Client End Session service, complete the following tasks:

1. Create an FTP Client End Session service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client End Session service. For information, see *Configuring the FTP Client End Session Service*.
3. Use the FTP Client End Session service in a business process.

## Configuring the FTP Client End Session Service

To configure the FTP Client End Session service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client End Session service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client End Session service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client End Session service:

| Parameter | Description |
|---|---|
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client End Session service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and the FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Business Process Example

The following example business process illustrates using the FTP Client End
Session service:

```
<sequence>
  [[Insert FTP Client Begin Session]]
   <operation name="FTP END SESSION SERVICE">
     <participant name="FTPClientEndSession"/>
     <output message="EndSessionRequest">
           <assign to="SessionToken"
           from="/ProcessData/FtpBeginSessionServiceResults/
SessionToken/text()">
           </assign>
     </output>
     <input message="inmsg">
         <assign to="FtpEndSessionResults" from="*"></assign>
     </input>
   </operation>
</sequence>
```

# Chapter 79. FTP Client GET Service

The following table provides an overview of the FTP Client GET service:

| System name | FTP Client GET Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service is used to retrieve one or more documents from a specified directory on the trading partner's FTP server. |
| Business usage | You would use this service to retrieve one or more documents from a trading partner and move them into Sterling Integrator when the FTP protocol is required as the transport mechanism. |
| Usage example | A Sterling Integrator business process is executed that must retrieve a specified file from the external trading partner. Sterling Integrator uses the FTP Client GET service, working through the FTP Client adapter, to retrieve the file from a specified directory on the trading partner system. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services: <br> • FTP Client adapter <br> • FTP Client Begin Session service <br> • FTP Client CD service <br> • FTP Client DELETE service <br> • FTP Client End Session service <br> • FTP Client LIST service <br> • FTP Client MOVE service <br> • FTP Client PUT service <br> • FTP Client PWD service <br> • FTP Client QUOTE service <br> • FTP Client SITE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is initiated from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values: <br> • 0 – Success <br> • 1 – Error |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test. To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console: <br><br> 1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**. <br><br> 2. Find FTP Server Adapter. <br><br> 3. If not already selected, select the **Enabled** check box. <br><br> To test this service, perform the following steps from the Sterling Integrator Admin Console: <br><br> 1. Choose **Business Processes** > **Manager**. <br><br> 2. Find the FTPClientDemoAllServices business process. <br><br> 3. Run the FTPClientDemoAllServices business process with the following settings: <br> • Run As User = Admin <br> • Server filename = install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml. <br><br> 4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule: <br> • FTP Client GET service – If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. <br> • FTP Client QUOTE service – This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client GET Service

To implement the FTP Client GET service, complete the following tasks:

1. Create an FTP Client GET service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.

2. Configure the FTP Client GET service. For information, see *Configuring the FTP Client GET Service*.

3. Use the FTP Client GET service in a business process.

## Configuring the FTP Client GET Service

To configure the FTP Client GET service, you must specify settings for the following fields in the UI or the GPM:

| Field | Description |
|---|---|
| Name | Name this adapter will have in Sterling Integrator |
| Description | Description of adapter |
| Select a Group | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br><br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br><br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |
| Config | Name of the service configuration. |
| CheckFileSize | Use to check that the file size is stable before downloading the file. Valid values are YES and NO. NO is the default value. When YES is specified, the FTP GET service checks the file size at 5-second intervals. The FTP GET service downloads the file only when the file size remains unchanged. Any change in the file size indicates that the file is still being transferred to the server and the service sends an error to the business process. Optional. You cannot use this parameter if RemoteFilePattern is specified.<br>**Note:** This parameter is supported with the following FTP servers:<br><br>• Windows IIS FTP server<br><br>• Standard UNIX FTP server – such as SUN Solaris, HP-UP, AIX<br><br>• Standard LINUX FTP server – Redhat<br><br>• War FTP Daemon 1.70/80 series (Windows) |
| ConnectionType | Value that describes which way the data connection will be made when the data is transferred. Optional. Value values are:<br><br>• ACTIVE — The server will make the connection. (Default)<br><br>• PASSIVE — The adapter will make the connection. |
| ListNamesErrorSetSuccess | Ignore 550 error code when executing the NLST command. Optional. Valid values are YES and NO. |

| Field | Description |
|---|---|
| RemoteFileName | The name of the file to be retrieved from the remote trading partner. Optional. You cannot use this parameter if RemoteFilePattern is specified.<br>**Note:** Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank. |
| RemoteFilePattern | The file filter pattern. Using this field activates multiple-get mode. Optional. You cannot use this parameter if RemoteFileName is specified.<br>**Note:** Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank. |
| RepresentationType | The FTP representation type that will be used for the transfer. Optional. Valid values are:<br>• ASCII – transfers the data in ASCII mode<br>• BINARY – transfers the data in binary mode (default) |
| ResponseTimeout | Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client GET service resets the value to 1 second. |
| RetrieveErrorSetSuccess | Ignore 550 error code when executing the RETR command. Optional. Valid values are YES and NO. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |
| DelayWaitingOnIO | Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service is operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |

## Parameters Passed from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client GET service:

| Parameter | Description |
|---|---|
| CheckFileSize | Use to check that the file size is stable before downloading the file. Valid values are YES and NO. NO is the default value. When YES is specified, the FTP GET service checks the file size at 5-second intervals. The FTP GET service downloads the file only when the file size remains unchanged. Any change in the file size indicates that the file is still being transferred to the server and the service sends an error to the business process. Optional. You cannot use this parameter if RemoteFilePattern is specified.<br>**Note:** This parameter is supported with the following FTP servers:<br>• Windows IIS FTP server<br>• Standard UNIX FTP server – such as SUN Solaris, HP-UP, AIX<br>• Standard LINUX FTP server – Redhat<br>• War FTP Daemon 1.70/80 series (Windows) |
| ConnectionType | Value that describes which way the data connection will be made when the data is transferred. Optional. Value values are:<br>• ACTIVE — The server will make the connection. (Default)<br>• PASSIVE — The adapter will make the connection. |
| ListNamesErrorSetSuccess | Ignore 550 error code when executing the NLST command. Optional. Valid values are YES and NO. |
| RemoteFileName | The name of the file to be retrieved from the remote trading partner. If a value is entered in this field, RemoteFilePattern cannot be used. Optional. You cannot use this parameter if RemoteFilePattern is specified.<br>**Note:** Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank. |
| RemoteFilePattern | The file filter pattern. Using this field activates multiple-get mode. If a value is entered in this field, then RemoteFileName cannot be used. Optional. You cannot use this parameter if RemoteFileName is specified.<br>**Note:** Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank. |
| RepresentationType | The FTP representation type that will be used for the transfer. Optional. Valid values are:<br>• ASCII – transfers the data in ASCII mode<br>• BINARY – transfers the data in binary mode (default) |
| ResponseTimeout | Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client GET service resets the value to 1 second. |
| RetrieveErrorSetSuccess | Ignore 550 error code when executing the RETR command. Optional. Valid values are YES and NO. |

| Parameter | Description |
|---|---|
| SaveTranscript | Indicates how to handle the transcript. Valid values are: <br>• erroronly – persists the transcript only when an error occurs <br>• on – always persists the transcript <br>Default is on. Optional. |
| DelayWaitingOnIO | Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service is operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. <br>**Note:** The session token is returned from the FTP Client Begin Session service. |

## Parameters Passed from Service to Business Process

The following table contains the parameters passed from the FTP Client GET service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required. |
| DocumentList | Provides a list of document IDs that were created for the files retrieved by the FTP Client GET service. Required. <br>**Note:** If a single document was retrieved, the service will place it as the primary document. |

## Business Process Examples

The following example business processes illustrate using commands supported by the FTP Client GET service.

This process gets a binary file named TestDoc using the passive connection type from the server:

```
<sequence>
  [[Insert FTP Client Begin Session here]]
    <operation name="FTP GET SERVICE">
      <participant name="FTPClientGet"/>
      <output message="GetRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="RemoteFileName">TestDoc</assign>
      <assign to="ConnectionType">PASSIVE</assign>
      <assign to="RepresentationType">BINARY</assign>
    </output>
    <input  message="inmsg">
      <assign to="FtpGetServiceResults" from="*"></assign>
```

```
      </input>
    </operation>
  [[Insert FTP Client End Session here]]
</sequence>
```

This process illustrates using a multiple GET command:

```
 <sequence>
  [[Insert FTP Client Begin Session here]]
    <operation name="FTP MULTIPLE GET SERVICE">
    <participant name="FTPClientGet"/>
    <output message="GetRequest">
       <assign to="SessionToken"
         from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="RepresentationType">BINARY</assign>
    </output>
    <input message="inmsg">
      <assign to="FtpGetServiceResults" from="*"></assign>
    </input>
  </operation>
  [[Insert FTP Client End Session here]]
</sequence>
```

The following example business process illustrates using an implicit assign to add
a message from the FTP Client GET service to the process data:

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

The following example business process illustrates using an explicit assign to add a
message from the FTP Client GET service to the process data:

```
<input message="inmsg">
   <assign to="StatusReport" from="Status_Rpt(&apos;StatusReport&apos;)">
       </assign>
   <assign to="FTPGetResults" from="*"><assign>
</input>
```

Sterling Integrator supports either implicit assign or explicit assign, but not both at
the same time, for example:

```
<input message="inmsg">
   <assign to="StatusReport" from="Status_Rpt(&apos;StatusReport&apos;)">
      </assign>
   <assign to="." from="*"></assign>
</input>
```

# Chapter 80. FTP Client LIST Service

The following table provides an overview of the FTP Client LIST service:

| System name | FTP Client LIST Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | Use this service to retrieve a list of documents in a specified directory on the trading partner's FTP server. |
| Business usage | 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 FTP protocol is required as the transport mechanism. |
| Usage example | A Sterling Integrator business process is executed that must retrieve a list of files from the external trading partner. Sterling Integrator uses the FTP Client LIST service, working through the FTP Client adapter, to retrieve the list of files from a specified directory on the trading partner system. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client SITE service<br>• FTP Client QUOTE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0 – Success<br>• 1 – Error |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.<br><br>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**.<br><br>2. Find FTP Server Adapter.<br><br>3. If not already selected, select the **Enabled** check box.<br><br>To test this service, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Manager**.<br><br>2. Find the FTPClientDemoAllServices business process.<br><br>3. Run the FTPClientDemoAllServices business process with the following settings:<br>• Run As User = Admin<br>• Server filename = install_dir/installed_data/ psftpclient/FTPClientDemoImport.xml.<br><br>4. Verify that the business process runs successfully.<br><br>Debug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. |

## Implementing the FTP Client LIST Service

To implement the FTP Client LIST service:

1. Create an FTP Client LIST service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client LIST service. For information, see *Configuring the FTP Client LIST Service*.
3. Use the FTP Client LIST service in a business process.

## Configuring the FTP Client LIST Service

To configure the FTP Client LIST service, you must specify settings for the following fields in the GPM:

| GPM Field | Description |
|---|---|
| Config | Name of the service configuration. |
| ConnectionType | Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are:<br>• ACTIVE - The server will make the connection. (Default)<br>• PASSIVE - The adapter will make the connection. |
| NamesOnly | Whether to include just names or all information to the server. Optional.<br>• YES – Causes the FTP protocol command NLST to be sent to the server. The results will be a stream of names of files and no other information.<br>• NO – Causes the FTP protocol command LIST to be sent to the server. The results will vary in form and information depending on the server. (Default) |
| RemoteFileName | File name or pattern to do the listing for. If not included the listing will be for a *.* pattern (depending on server implementation). Optional. |
| RepresentationType | Representation used for transferring the file. Optional. Valid values are:<br>• ASCII – Transfers the data in ASCII mode<br>• BINARY – Transfers the data in binary mode (default) |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client LIST service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript (Default) |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client LIST service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |

| Parameter | Description |
|---|---|
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required. |
| ListNames | Provides information about the files included in the specified directory. Optional. Only returned if NamesOnly=YES. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client LIST service:

| Parameter | Description |
|---|---|
| ConnectionType | Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are:<br>• ACTIVE - The server will make the connection. (Default)<br>• PASSIVE - The adapter will make the connection. |
| NamesOnly | Whether to include just names or all information to the server. Optional.<br>• YES – Causes the FTP protocol command NLST to be sent to the server. The results will be a stream of names of files and no other information.<br>• NO – Causes the FTP protocol command LIST to be sent to the server. The results will vary in form and information depending on the server. (Default) |
| ListNamesErrorSetSuccess | Ignore 550 error code when executing the NLST command. Optional. Valid values are:<br>• YES<br>• NO |
| RemoteFileName | File name or pattern to do the listing for. If not included the listing will be for a *.* pattern (depending on server implementation). Optional. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. The minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client LIST service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript (Default) |

## Business Process Example

The following example business processes illustrate the use of the FTP Client LIST service:

Example of a Names Only listing:

```
<sequence>
  [[ Insert FTP Client Begin Session ]]
    <operation name="FTP LIST SERVICE NAME">
    <participant name="FTPClientList"/>
    <output message="ListRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="NamesOnly">YES</assign>
      <assign to="RemoteFileName">*.txt</assign>
      <assign to="ConnectionType">PASSIVE</assign>
    </output>
    <input message="inmsg">
      <assign to="NameListResults" from="*"></assign>
    </input>
    </operation>
  [[ Insert FTP Client End Session ]]
</sequence>
```

Example of a Raw listing:

```
<sequence>
  [[ Insert FTP Client Begin Session ]]
    <operation name="FTP LIST SERVICE RAW">
    <participant name="FTPClientList"/>
    <output message="ListRequest">
      <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="NamesOnly">NO</assign>
      <assign to="ConnectionType">PASSIVE</assign>
    </output>
    <input message="inmsg">
      <assign to="RawListResults" from="*"></assign>
    </input>
    </operation>
  [[ Insert FTP Client End Session ]]
</sequence>
```

# Chapter 81. FTP Client MKD Service

The FTP Client MKD Service is used to create a directory on a trading partner's FTP server. The following table provides an overview of the FTP Client MKD service:

| System name | FTP Client MKD Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | Used to create a directory on the trading partner's FTP server. |
| Business usage | A business user would use this service to create directories and put the files into the respective directories on the server side when the FTP protocol is required as the transport mechanism. |
| Usage example | Your company uses a business process that translates documents that must be sent to a trading partner on a daily basis. After the translation, the Sterling Integrator uses the FTP Client MKD service, working through the FTP Client adapter, to create the directory with the current date so that the documents are placed in the specified directory on the trading partner system. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The FTP Client MKD service must be placed between an FTP Begin Session service and an FTP End Session service in a business process. You can use the CD service to change to the directory created by this service. Documents can be put or retrieved from the newly created directory using the PUT service and GET service. Related services are:<br><br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client GET service<br>• FTP Client LIST service<br>• FTP Client MOVE service<br>• FTP Client PWD service<br>• FTP Client PUT service<br>• FTP Client QUOTE service<br>• FTP Client SITE service<br>• FTP Client RMD service<br>• FTP Client MODE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |

| Business process context considerations | Business processes using the FTP client services should not be marked as Auto Resume. The nature of these protocol services requires an established session, which no longer exists in a restart situation. |
|---|---|
| Returned status values | Status values are:<br>• 0 – Success<br>• 1 – Error |
| Restrictions | None |
| Persistence level | Default |
| Testing considerations | Debug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (4xx or 5xx) then the business process produces a fault. If the error code is expected, use a BPML activity to continue interacting with the trading partner. There are two exceptions to this rule:<br>• FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document.<br>• FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Configuring the FTP Client MKD Service

To configure the FTP Client MKD service, you must specify settings for the following fields in the Admin console and in the GPM:

| UI Field | 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:<br>• None – Do not include the configuration in a service group at this time.<br>• 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.)<br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |

| UI Field | GPM Field | Description |
|---|---|---|
| | Config | Name of the service configuration. |
| | RemoteDirName | The directory name to be created on the remote system. Required. |
| | ResponseTimeout | The maximum number of seconds where no data transfer over the data channel is allowed before the service times out and raises a timeout exception. Optional. Valid value is any numeric value. If no value is specified or a timeout value of less than 30 is specified, Response Timeout is defaulted to 30 seconds. |
| | SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• Error only – Persists the transcript only when an error occurs.<br>• On – Always persists the transcript. Default is on. |
| | SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. **Note:** The session token is returned from the FTP Client Begin Session service. |

## Parameters Passed from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client MKD service:

| Parameter | Description |
|---|---|
| RemoteDirName | The directory name to be created on the remote system. Required. |
| ResponseTimeout | The maximum number of seconds where no data transfer over the data channel is allowed before the service times out and raises a timeout exception. Optional. Valid value is any numeric value. If no value is specified or a timeout value of less than 30 is specified, Response Timeout is defaulted to 30 seconds. |
| SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• Error only – Persists the transcript only when an error occurs.<br>• On – Always persists the transcript. Default is on. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |

## Parameters Passed from Service to Business Process

The following table contains the parameters passed from the FTP Client MKD service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. |

## Business Process Examples

### Example 1

The following business process uses the FTP Client MKD service to create a directory on the remote FTP server.

```
<process name="FtpExample">
  <sequence>
[[Insert Begin Session ]]
<operation name="FTP Client MKD Service">
      <participant name="FTPClientMkd"/>
      <output message="FTPClientMkdServiceTypeInputMessage">
        <assign to="." from="*"></assign>
        <assign to="RemoteDirName">FTPTEST</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
 [[Insert end session here]]
  </sequence>
</process>
```

### Example 2

The following business process uses the FTP Client MKD service to create a directory on the remote FTP server, change to the directory and put the primary document from Sterling Integrator on the remote FTP server.

```
<process name="FtpExample">
  <sequence>
[[Insert Begin Session ]]
<operation name="FTP Client MKD Service">
      <participant name="FTPClientMkd"/>
      <output message=" MkdRequest ">
        <assign to="SessionToken" from="/ProcessData/BeginSessionResults/
                SessionToken/text()"></assign>
        <assign to="RemoteDirName">NEWFTP1 </assign>
      </output>
      <input message="FtpMkdResults ">
 <assign to="MkdResults" from="*"/>
          </input>
  </operation>
<operation name="FTP Client CD Service">
      <participant name="FTPClientCd"/>
      <output message=" CdRequest ">
        <assign to="SessionToken" from="/ProcessData/BeginSessionResults/
                SessionToken/text()"></assign>
        <assign to="Directory">NEWFTP1</assign>
      </output>
```

```
      <input message=" FtpCdResults ">
        <assign to=" CdResults " from="*"></assign>
      </input>
    </operation>
<operation name="FTP Client PUT Service">
      <participant name="FTPClientPut"/>
      <output message=" PutRequest">
        <assign to="SessionToken" from="/ProcessData/BeginSessionResults/
                SessionToken/text()"></assign>
        <assign to="RemoteFileName">TestDoc</assign>
        <assign to="ConnectionType">PASSIVE</assign>
        <assign to="RepresentationType">ASCII</assign></output>
      </output>
     <input message="FtpPutResults">
        <assign to="PutResults" from="*"></assign>
      </input>
        <assign to="." from="*"></assign>
      </input>
    </operation>
[[Insert End Session]]
  </sequence>
</process>
```

# Chapter 82. FTP Client MODE Service

The FTP Client MODE service specifies the transmission mode used to send or receive data from the trading partner's FTP server. The following table provides an overview of the FTP Client MODE service:

| System name | FTP Client MODE Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | Used to specify the transmission mode used to send or receive data from the trading partner's FTP server. |
| Business usage | Enables you to send data in normal streamed mode or in deflated mode. Sending data in deflated mode minimizes the number of bytes transferred, compared to normal mode. |
| Usage example | Your company uses a business process that sends large documents to a trading partner. You use deflated mode to send the data. The data is compressed during the sending, which can increase the speed of the data transfer. The data is uncompressed after it is received by the FTP server. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The FTP Client MODE service must be placed in between an FTP Begin Session service and an FTP End Session service in a business process. Related services are:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client GET service<br>• FTP Client LIST service<br>• FTP Client MOVE service<br>• FTP Client PWD service<br>• FTP Client PUT service<br>• FTP Client QUOTE service<br>• FTP Client SITE service<br>• FTP Client MKD service<br>• FTP Client RMD service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | Business processes using the FTP client services should not be marked as Auto Resume. These protocol services require an established session, which will no longer exist in a re-start situation. |

| Returned status values | Status values are: |
|---|---|
| | • 0 – Success |
| | • 1 – Error |
| Restrictions | None |
| Persistence level | Default |
| Testing considerations | Debug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this error code is defined by the FTP specification (4xx or 5xx) then the business process produces a fault. If the error code is expected, use a BPML activity in your business process to continue interacting with the trading partner. There are two exceptions to this rule: |
| | • FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. |
| | • FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Configuring the FTP Client MODE Service

To configure the FTP Client MODE service, you must specify settings for the following fields in the Admin Console and in the GPM:

| UI Field | 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: |
| | | • 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. |
| | Config | Name of the service configuration. |

| UI Field | GPM Field | Description |
|---|---|---|
| | ResponseTimeout | The maximum number of seconds where no data transfer over the data channel is allowed before the service times out and raises a timeout exception. Optional. Valid value is any numeric value. If no value specified or a timeout value of less than 30 is specified, ResponseTime defaults to 30 seconds. |
| | SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• Error only – Persists the transcript only when an error occurs.<br>• On – Always persists the transcript. Default. |
| | SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. **Note:** The session token is returned from the FTP Client Begin Session service. |
| | TransmissionMode | How the data is sent. Required. Valid values are:<br>• Streamed (Default)<br>• Deflated |

The FTP session remains in the selected TranmissionMode until the mode service is used to change it to another mode.

## Parameters Passed from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client MODE service:

| Parameter | Description |
|---|---|
| TransmissionMode | How the data is sent. Required. Valid values are:<br>• Streamed<br>• Deflated (Default) |
| ResponseTimeout | The maximum number of seconds where no data transfer over the data channel is allowed before the service times out and raises a timeout exception. Optional. Valid value is any numeric value. If no value specified or a timeout value of less than 30 is specified, it will be defaulted to 30 seconds. |
| SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• Error only – Persists the transcript only when an error occurs.<br>• On – Always persists the transcript. Default. |

| Parameter | Description |
|---|---|
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |

## Parameters Passed from Service to Business Process

The following table contains the parameters passed from the FTP Client MODE service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. |

## Business Process Examples

### Example 1

The following business process uses the FTP Client MODE service to transfer the data in deflated mode:

```
<process name="FtpExample">
  <sequence>
[[Insert Begin Session ]]
<operation name="FTP Client MODE Service">
      <participant name=" FTPClientMode"/>
      <output message=" ModeRequest ">
          <assign to="TransmissionMode">Z</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
 [[Insert end session here]]
   </sequence>
</process>
```

### Example 2

The following business process uses the FTP Client MODE service to put a document on the remote FTP server in the deflated mode.

```
<process name="FtpExample">
  <sequence>
[[Insert Begin Session ]]
<operation name="FTP Client MODE Service">
      <participant name="FTPClientMode"/>
      <output message=" ModeRequest ">
        <assign to="SessionToken" from="/ProcessData/BeginSessionResults/
                  SessionToken/text()"></assign>
        <assign to="TransmissionMode">Z</assign>
      </output>
      <input message="FtpModeResults ">
 <assign to="ModeResults" from="*"/>
          </input>
    </operation>
```

```
<operation name="FTP Client PUT Service">
      <participant name="FTPClientPut"/>
      <output message=" PutRequest">
        <assign to="SessionToken" from="/ProcessData/BeginSessionResults/
                 SessionToken/text()"></assign>
        <assign to="RemoteFileName">TestDoc</assign>
        <assign to="ConnectionType">PASSIVE</assign>
        <assign to="RepresentationType">ASCII</assign></output>
      </output>
     <input message="FtpPutResults">
        <assign to="PutResults" from="*"></assign>
     </input>
        <assign to="." from="*"></assign>
     </input>
    </operation>
[[Insert End Session]]
  </sequence>
</process>
```

**Example 3**

The following business process uses the FTP Client MODE service to retrieve a
document from the remote FTP server to Sterling Integrator in the deflated mode.

```
<process name="FtpExample">
  <sequence>
[[Insert Begin Session ]]
<operation name="FTP Client MODE Service">
      <participant name="FTPClientMode"/>
      <output message=" ModeRequest ">
        <assign to="SessionToken" from="/ProcessData/BeginSessionResults/
                 SessionToken/text()"></assign>
        <assign to="TransmissionMode">Z</assign>
      </output>
      <input message="FtpModeResults ">
 <assign to="ModeResults" from="*"/>
            </input>
  </operation>
<operation name="FTP Client GET Service">
      <participant name="FTPClientGet"/>
      <output message=" GetRequest">
        <assign to="SessionToken" from="/ProcessData/BeginSessionResults/
                 SessionToken/text()"></assign>
        <assign to="RemoteFileName">TestDoc</assign>
        <assign to="ConnectionType">PASSIVE</assign>
        <assign to="RepresentationType">ASCII</assign></output>
      </output>
     <input message="FtpPutResults">
        <assign to="PutResults" from="*"></assign>
     </input>
        <assign to="." from="*"></assign>
     </input>
    </operation>
[[Insert End Session]]
  </sequence>
</process>
```

# Chapter 83. FTP Client MOVE Service

The following table provides an overview of the FTP Client MOVE service:

| | |
|---|---|
| System name | FTP Client MOVE Service |
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service is used to rename a document or move it from one directory to another. |
| Business usage | You would use this service to rename or move a document on a trading partner's system when the FTP protocol is required as the communication mechanism with the trading partner. |
| Usage example | A Sterling Integrator business process is executed that requires moving a document on the trading partner's system. Sterling Integrator uses the FTP Client MOVE service, working through the FTP Client adapter, to move the specified document from one directory to another on the trading partner system. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services: <br> • FTP Client adapter <br> • FTP Client Begin Session service <br> • FTP Client CD service <br> • FTP Client DELETE service <br> • FTP Client End Session service <br> • FTP Client LIST service <br> • FTP Client GET service <br> • FTP Client PUT service <br> • FTP Client PWD service <br> • FTP Client SITE service <br> • FTP Client QUOTE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values: <br> • 0 –Success <br> • 1 –Error |
| Restrictions | No |
| Persistence level | Default |

| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test. |
|---|---|
| | To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**. |
| | 2. Find FTP Server Adapter. |
| | 3. If not already selected, select the **Enabled** check box. |
| | To test this service, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Manager**. |
| | 2. Find the FTPClientDemoAllServices business process. |
| | 3. Run the FTPClientDemoAllServices business process with the following settings: |
| | • Run As User = Admin |
| | • Server filename = <install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml. |
| | 4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule: |
| | • FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. |
| | • FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client MOVE Service

To implement the FTP Client MOVE service, complete the following tasks:

1. Create an FTP Client MOVE service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client MOVE service. For information, see *Configuring the FTP Client MOVE Service*.
3. Use the FTP Client MOVE service in a business process.

## Configuring the FTP Client MOVE Service

To configure the FTP Client MOVE service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| RemoteFromFileName | The current name of the remote file. Required. |
| RemoteToFileName | The new name of the remote file. Required. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client MOVE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. **Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client MOVE service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client MOVE service:

| Parameter | Description |
|---|---|
| RemoteFromFileName | The current name of the remote file. Required. |
| RemoteToFileName | The new name of the remote file. Required. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client MOVE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Business Process Example

The following example business process illustrates using the FTP Client MOVE service:

```
<sequence>
  [[Insert FTP Client Begin Session ]]
    <operation name="FTP MOVE SERVICE">
    <participant name="FTPClientMove"/>
    <output message="MoveRequest">
      <assign to="SessionToken"
          from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
      </assign>
      <assign to="RemoteFromFileName">oldFileNameAndDirectory</assign>
      <assign to="RemoteToFileName">newFileNameAndDirectory</assign>
    </output>
    <input message="inmsg">
      <assign to="FtpMoveResults" from="*"></assign>
    </input>
    </operation>
  [[Insert FTP Client End Session]]
</sequence>
```

# Chapter 84. FTP Client PUT Service

The FTP Client PUT service places a document or documents in a specified directory on the trading partner's FTP server. The following table provides an overview of the FTP Client PUT service:

| System name | FTP Client PUT Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | Used to place a document or documents in a specified directory on the trading partner's FTP server. |
| Business usage | Use this service to transfer a document or documents from Sterling Integrator to a trading partner when the FTP protocol is required as the transport. |
| Usage example | You run a business process that translates a document that must be sent to a trading partner. After the translation, Sterling Integrator uses the FTP Client PUT service, working through the FTP Client adapter, to place the document in a specified directory on the trading partner system. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client GET service<br>• FTP Client LIST service<br>• FTP Client MOVE service<br>• FTP Client PWD service<br>• FTP Client QUOTE service<br>• FTP Client SITE service<br>• FTP Client MKD service<br>• FTP Client RMD service<br>• FTP Client MODE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | Business processes that use the FTP client services should not be marked as Auto Resume. These protocol services require an established session, which will no longer exist in a re-start situation. |
| Returned status values | Returned status values:<br>• 0 – Success<br>• 1 – Error |

| Restrictions | None |
|---|---|
| Persistence level | Default |
| Testing considerations | Debug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this error code is defined by the FTP specification (4xx or 5xx) then the business process produces a fault. If the error code is expected, use a BPML activity to continue interacting with the trading partner. There are two exceptions to this rule:<br>• FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document.<br>• FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP PUT Client Service

To implement the FTP Client PUT service, complete the following tasks:

1. Create an FTP Client PUT service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client PUT service. For information, see *Configuring the FTP Client PUT Service*.
3. Use the FTP Client PUT service in a business process.

## Configuring the FTP Client PUT Service

To configure the FTP Client PUT service, you must specify settings for the following fields in the UI or the GPM:

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

| UI Field | GPM | Description |
|---|---|---|
| Select a Group | | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br><br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br><br>• 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. |
| | Config | Name of the service configuration. |
| | ConnectionType | Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are:<br><br>• ACTIVE - The server will make the connection. Default.<br><br>• PASSIVE - The adapter will make the connection. |
| | DelayWaitingOnIO | Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional. |
| | DocumentId | Document ID to PUT to the remote server. A single DocumentId can appear directly in the message to the service or any number of DocumentIds can appear under the DocumentList element. Optional. **Note:** The FTP Client PUT service will use DocumentList if a list is provided. If no list is specified in DocumentList, the service will use DocumentId. The service will not use both DocumentList and DocumentId. If no values are specified for either DocumentList or DocumentId, the service will PUT the primary document to the remote server. |
| | DocumentList | List of documents to PUT to the remote server. Each item must be a DocumentId. A list could look like the following example:<br><br>`<DocumentList>`<br>`<DocumentId>12345</DocumentId>`<br>`<DocumentId>67890</DocumentId>`<br>`</DocumentList>` |

| UI Field | GPM | Description |
|---|---|---|
| | RemoteFileName | Filename that Sterling Integrator uses to place the document on the remote system. If not specified, the name of the document will be used. Do not use this parameter if putting multiple documents on a server. Optional. |
| | RepresentationType | Representation used for transferring the file. Optional. Valid values:<br>• ASCII – Transfers the data in ASCII mode<br>• BINARY – Transfers the data in binary mode (Default)<br>• EBCDIC – Transfers the data in EBCDIC mode<br>• L 8 – Transfers the data in Local Byte 8 mode that is implicitly Binary mode |
| | ResponseTimeout | Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates. Optional. Minimum value is 1 second. Valid value is any numeric value of 1 or higher. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. If the value is less than 1 second, the FTP Client PUT service resets the value to 1 second. |
| | SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• erroronly – Persists the transcript only when an error occurs<br>• on – Always persists the transcript (Default) |
| | SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| | StoreAction | Specifies the type of store that must be done. Optional. Valid values are:<br>• Normal - Puts a new file with the specified name. Default.<br>• Append - Appends to the file if it already exists. Otherwise puts a new file with the specified name.<br>• Unique - Saves the file with a unique name if a file with the same name already exists. |

| UI Field | GPM | Description |
|---|---|---|
| | UseDocBodyName | Specifies whether to use document body name as the remote file name. This parameter is only use in MPUT operation. Valid values are:<br>• YES – Use document body name<br>• NO – Use document name<br>Default is NO. Optional. |

## Parameters Passed from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client PUT service:

| Parameter | Description |
|---|---|
| ConnectionType | Value that describes which way the data connection will be made when the data is transferred. Optional. Valid values are:<br>• ACTIVE -The server will make the connection. Default.<br>• PASSIVE - The adapter will make the connection. |
| DocumentId | Document ID to PUT to the remote server. A single DocumentId can appear directly in the message to the service or any number of DocumentIds can appear under the DocumentList element. Optional.<br>**Note:** The FTP Client PUT service will use DocumentList if a list is provided. If no list is specified in DocumentList, the service will use DocumentId. The service will not use both DocumentList and DocumentId. If no values are specified for either DocumentList or DocumentId, the service will PUT the primary document to the remote server. |
| DocumentList | List of documents to PUT to the remote server. Each item must be a DocumentId. A list could look like the following example:<br>`<DocumentList>`<br>`  <DocumentId>12345</DocumentId>`<br>`  <DocumentId>67890</DocumentId>`<br>`</DocumentList>` |
| RemoteFileName | Filename that Sterling Integrator uses to place the document on the remote system. If not specified, the name of the document will be used. Do not use this parameter if putting multiple documents on a server. Optional. |
| RepresentationType | Representation used for transferring the file. Optional. Valid values:<br>• ASCII – Transfers the data in ASCII mode<br>• BINARY – Transfers the data in binary mode (default) |

| Parameter | Description |
|---|---|
| ResponseTimeout | Maximum number of seconds of inactivity during data transfer between the FTP Client and the FTP Server. The FTP Client waits during the data transfer, before the session times out and terminates. Optional. Minimum value is 1 second. Valid value is any numeric value of 1 or higher. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. If the value is less than 1 second, the FTP Client PUT service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| StoreAction | Specifies the type of store that must be done. Optional. Valid values are:<br>• Normal - Puts a new file with the specified name. Default.<br>• Append - Appends to the file if it already exists. Otherwise puts a new file with the specified name.<br>• Unique - Saves the file with a unique name if a file with the same name already exists. |
| UseDocBodyName | Specifies whether to use document body name as the remote file name. This parameter is only use in MPUT operation. Optional. Valid values are:<br>• YES – Use document body name<br>• NO – Use document name (Default) |
| DelayWaitingOnIO | Specifies the number of seconds to wait for the data transfer to complete before going into WAITING_ON_IO state. If -1 is specified, the service is operates in blocking mode. It will wait until the data transfer has completed. Valid value is any numerical value. Optional. |
| SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br>• erroronly – Persists the transcript only when an error occurs<br>• on – Always persists the transcript (Default) |

## Parameters Passed from Service to Business Process

The following table contains the parameters passed from the FTP Client PUT service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. |

## Business Process Examples

### Example 1

The following business process uses the FTP Client adapter to send the primary document from from Sterling Integrator to the remote FTP server:

```
<process name="FtpExample">
<sequence>
[[Insert Begin Session ]]
  <operation name="FTP PUT SERVICE">
  <participant name="FTPClientPut"/>
    <output message="PutRequest">
        <assign to="SessionToken"
from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
</assign>
        <assign to="RemoteFileName">TestDoc</assign>
        <assign to="ConnectionType">PASSIVE</assign>
        <assign to="RepresentationType">BINARY</assign>
        <assign to="." From="PrimaryDocument"></assign>
    </output>
    <input message="inmsg">
        <assign to="FtpPutServiceResults" from="*"></assign>
    </input>
 </operation>
</sequence>
[[Insert end session here]]
</process>
```

### Example 2

The following business process uses the FTP Client PUT service to send all documents received from a GET from Application to the remote FTP server.

```
<process name="FTP_PUT">
<sequence name="optional">
[[Insert begin session here?]]
    <operation name="Get">
      <participant name="FTPClientGet"/>
      <output message="GetRequest">
          <assign to="SessionToken"
from="/ProcessData/BeginSessionResults/SessionToken/text()"></assign>
          <assign to="RemoteFilePattern">*.*</assign>
      </output>
      <input message="GetResults">
            <assign to="GetResults" from="DocumentList"/>
        </input>
    </operation>
     <operation name="Put">
       <participant name="FTPClientPut"/>
         <output message="PutRequest">
           <assign to="SessionToken"
from="/ProcessData/BeginSessionResults/SessionToken/text()"></assign>
            <assign to="." From="/ProcessData/GetResults/DocumentList"/>
            <assign to="RepresentationType">ASCII</assign> </output>
          <input message="FtpPutResults">
            <assign to="PutResults" from="*"></assign>
          </input>
          </operation>
[[Insert End Session]]
     </sequence>
</process>
```

### Example 3

The following business process uses the FTP Client PUT service to send all documents in a DocumentList from Application to the remote FTP server.

```
<process name="FTP_PUT">
  <sequence name="optional">
[[Insert begin session here?]]
    <operation name="Put">
        <participant name="FTPClientPut"/>
        <output message="PutRequest">
          <assign to="SessionToken"
from="/ProcessData/BeginSessionResults/SessionToken/text()"></assign>
          <assign to="DocumentList/DocumentId" append="true">
xxxxxxfffff
</assign>
          <assign to="DocumentList/DocumentId" append="true">
yyyyyyfffff
</assign>
          <assign to="RepresentationType">ASCII</assign>
        </output>
        <input message="FtpPutResults">
           <assign to="PutResults" from="*"></assign>
        </input>
    </operation>
[[Insert End Session]]
    </sequence>
</process>
```

# Chapter 85. FTP Client PWD Service

The following table provides an overview of the FTP Client PWD service:

| System name | FTP Client PWD Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service is used to print the working directory on the trading partner's FTP server. |
| Business usage | You would use this service to get information about the current working directory on the trading partner FTP server. |
| Usage example | A Sterling Integrator business process is executed that places a document in a specific directory on the trading partner's system. The business process writer wants to ensure that status information within the business process includes the name of the directory where the document was placed. The business process writer includes the FTP Client PWD service in the process definition and the service places the directory information in the business process. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client LIST service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client SITE service<br>• FTP Client QUOTE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0 –Success<br>• 1 –Error |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.<br><br>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**.<br><br>2. Find FTP Server Adapter.<br><br>3. If not already selected, select the **Enabled** check box.<br><br>To test this service, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Manager**.<br><br>2. Find the FTPClientDemoAllServices business process.<br><br>3. Run the FTPClientDemoAllServices business process with the following settings:<br>• Run As User = Admin<br>• Server filename = <install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml.<br><br>4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:<br>• FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document.<br>• FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client PWD Service

To implement the FTP Client PWD service, complete the following tasks:

1. Create an FTP Client PWD service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client PWD service. For information, see *Configuring the FTP Client PWD Service*.
3. Use the FTP Client PWD service in a business process.

## Configuring the FTP Client PWD Service

To configure the FTP Client PWD service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client PWD service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client PWD service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. The text includes the server-specific text indicating the current working directory for the session. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client PWD service:

| Parameter | Description |
|---|---|
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client PWD service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Business Process Example

The following example business process illustrates using the FTP Client PWD service:

```
<sequence>
  [[ Insert FTP Client Begin Session ]]
    <operation name="FTP PWD SERVICE">
    <participant name="FTPClientPwd"/>
    <output message="PwdRequest">
       <assign to="SessionToken"
         from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
       </assign>
    </output>
    <input  message="inmsg">
       <assign to="FTPClientPwdResults" from="*"></assign>
    </input>
  </operation>
  [[ Insert FTP Client End Session ]]
</sequence>
```

# Chapter 86. FTP Client RMD Service

The FTP Client RMD service removes a specific directory on the trading partner's FTP server. The following table provides an overview of the FTP Client RMD service:

| | |
|---|---|
| System name | FTP Client RMD Service |
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service removes a specific directory on the trading partner's FTP server. |
| Business usage | A business user would use this service to remove desired directories on the server side when the FTP protocol is required as the transport mechanism. |
| Usage example | A business process is executed that translates documents that must be sent to a trading partner on a daily basis and places them in the directory named for the current date. Once the documents are processed, a business process is executed that removes the directory. This prevents an accumulation of unused and obsolete directories on the server. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The FTP Client RMD service must be placed between an FTP Begin Session service and an FTP End Session service in a business process. Related services are:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client GET service<br>• FTP Client LIST service<br>• FTP Client MOVE service<br>• FTP Client PWD service<br>• FTP Client PUT service<br>• FTP Client QUOTE service<br>• FTP Client SITE service<br>• FTP Client MODE service<br>• FTP Client MKD service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | Business processes using the FTP client services should not be marked as Auto Resume. The nature of these protocol services requires an established session, which will no longer exist in a re-start situation. |

| Returned status values | • 0 – Success |
| | • 1 – Error |
| Restrictions | None |
| Persistence level | Default |
| Testing considerations | Debug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (4xx or 5xx) then the business process will produce a fault. If the error code is expected, use a BPML activity to continue interacting with the trading partner. There are two exceptions to this rule: |
| | • FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. |
| | • FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client RMD Service

To implement the FTP Client RMD service, complete the following tasks:

1. Create an FTP Client RMD service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed).
2. Configure the FTP Client RMD service.
3. Use the FTP Client RMD service in a business process.

## Configuring the FTP Client RMD Service

To configure the FTP Client RMD service, you must specify settings for the following fields in the Admin Console and in the GPM:

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

| UI Field | GPM Field | Description |
|---|---|---|
| Select a group | | Select one of the options:<br><br>• None – Do not include the configuration in a service group at this time.<br><br>• 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.)<br><br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |
| | Config | Name of the service configuration. |
| | RemoteDirName | The directory name to be removed from the remote system. Required. |
| | ResponseTimeout | The maximum number of seconds where no data transfer over the data channel is allowed before the service times out and raises a timeout exception. Optional. Valid value is any numeric value. If no value specified or a timeout value of less than 30 is specified, ResponseTimeout defaults to 30 seconds. |
| | SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br><br>• Error only – Persists the transcript only when an error occurs<br><br>• On – Always persists the transcript (Default) |
| | SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |

## Parameters Passed from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client RMD service:

| Parameter | Description |
|---|---|
| RemoteDirName | The directory name to be removed from the remote system. Required. |
| ResponseTimeout | The maximum number of seconds where no data transfer over the data channel is allowed before the service times out and raises a timeout exception. Optional. Valid value is any numeric value. If no value specified or a timeout value of less than 30 is specified, it will be defaulted to 30 seconds. |

| Parameter | Description |
|---|---|
| SaveTranscript | Indicates how to handle the transcript. Optional. Valid values are:<br><br>• Error only – Persists the transcript only when an error occurs<br>• On – Always persists the transcript (Default) |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |

## Parameters Passed from Service to Business Process

The following table contains the parameters passed from the FTP Client RMD service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. |

## Business Process Example

The following business process uses the FTP Client RMD service to remove a directory on the remote FTP server.

```
<process name="FtpExample">
  <sequence>
[[Insert Begin Session ]]
<operation name="FTP Client RMD Service">
      <participant name="FTPClientRmd"/>
      <output message=" RmdRequest ">
               <assign to="." from="*"></assign>
               <assign to="RemoteDirName">FTPTEST</assign>
      </output>
      <input message="FtpRmdResults ">
 <assign to="RmdResults" from="*"/>
      </input>
  </operation>
    </operation>
 [[Insert end session here]]
  </sequence>
</process>
```

# Chapter 87. FTP Client QUOTE Service

The following table provides an overview of the FTP Client QUOTE service:

| System name | FTP Client QUOTE Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | The FTP Client QUOTE service is used to pass custom commands to the trading partner's FTP server. |
| Business usage | You would use this service to pass commands that the FTP server can support, but which are not included in the standard protocol. |
| Usage example | A Sterling Integrator business process is executed that translates a document that must be sent to a trading partner. After the translation, Sterling Integrator establishes a session with the trading partner FTP server (using the FTP Client adapter), and uses the QUOTE service to issue a print command on the host. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client LIST service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client SITE service |
| Application requirements | An FTP server at the external trading partner location. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0 – Success<br>• 1 – Error |
| Restrictions | None |
| Persistence level | Default |

| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test. |
|---|---|
| | To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**. |
| | 2. Find FTP Server Adapter. |
| | 3. If not already selected, select the **Enabled** check box. |
| | To test this service, perform the following steps from the Sterling Integrator Admin Console: |
| | 1. Choose **Business Processes** > **Manager**. |
| | 2. Find the FTPClientDemoAllServices business process. |
| | 3. Run the FTPClientDemoAllServices business process with the following settings: |
| | • Run As User = Admin |
| | • Server filename = <install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml. |
| | 4. Verify that the business process runs successfully. Debug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule: |
| | • FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document. |
| | • FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client QUOTE Service

To implement the FTP Client QUOTE service, complete the following tasks:

1. Create an FTP Client QUOTE service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client QUOTE service. For information, see *Configuring the FTP Client QUOTE Service*.
3. Use the FTP Client QUOTE service in a business process.

## Configuring the FTP Client QUOTE Service

To configure the FTP Client QUOTE service, you must specify settings for the following fields in the GPM:

| Field | Description |
|-------|-------------|
| Config | Name of the service configuration. |
| QuoteCommand | Enter the exact text of the command that will be run on the server (do not include the word 'Quote'). Required. Valid value is any text to send to the server. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client QUOTE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client QUOTE service to the business process:

| Parameter | Description |
|-----------|-------------|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. If the exchange involves a waiting state, the final transcript document ID contains the record of the full exchange. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client QUOTE service:

| Parameter | Description |
|---|---|
| QuoteCommand | Enter the exact text of the command that will be run on the server (do not include the word 'Quote'). Required. Valid value is any text to send to the server. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client QUOTE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required. **Note:** The session token is returned from the FTP Client Begin Session service. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are: <br> • erroronly – persists the transcript only when an error occurs <br> • on – always persists the transcript <br><br> Default is on. Optional. |

## Business Process Example

The following example business process illustrates sending a site command using the FTP Client QUOTE service:

```
 <sequence>
  [[ Insert FTP Client Begin Session ]]
    <operation name="FTP QUOTE SERVICE">
    <participant name="FTPClientQuote"/>
    <output message="QuoteRequest">
      <assign to="SessionToken" △
        from="/ProcessData/FtpBeginSessionServiceResults/
SessionToken/text()">
      </assign>
      <assign to="QuoteCommand">SITE HELP</assign>
    </output>
    <input message="inmsg">
      <assign to="FTPClientQuoteResults" from="*"></assign>
    </input>
    </operation>
  [[  Insert FTP Client Begin Session ]]
</sequence>
```

# Chapter 88. FTP Client SITE Service

The following table provides an overview of the FTP Client SITE service:

| | |
|---|---|
| System name | FTP Client SITE Service |
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > FTP Client |
| Description | This service is used to send site-specific control commands to an FTP server. |
| Business usage | You would use this service to send site-specific control commands to a trading partner FTP server. |
| Usage example | A Sterling Integrator business process is executed that must send a site-specific control command to a trading partner. Sterling Integrator uses the FTP Client SITE service to send a site-specific control command to the trading partner's FTP server. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• FTP Client adapter<br>• FTP Client Begin Session service<br>• FTP Client CD service<br>• FTP Client DELETE service<br>• FTP Client End Session service<br>• FTP Client LIST service<br>• FTP Client GET service<br>• FTP Client MOVE service<br>• FTP Client PUT service<br>• FTP Client PWD service<br>• FTP Client QUOTE service |
| Application requirements | An FTP server at the external trading partner location. Information about this server must be configured in the trading profile. |
| Initiates business processes? | This service does not initiate business processes. |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0 –Success<br>• 1 –Error |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | Test this service by running the FTPClientDemoAllServices business process provided with Sterling Integrator. This business process tests the FTP Client adapter and all its related services. The FTPClientDemoAllServices business process uses the preconfigured instance of the FTP Server adapter, which is disabled by default, and must be enabled before running this test.<br><br>To verify that the preconfigured FTP Server adapter is enabled, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Deployment** > **Services** > **Configuration**.<br><br>2. Find FTP Server Adapter.<br><br>3. If not already selected, select the **Enabled** check box.<br><br>To test this service, perform the following steps from the Sterling Integrator Admin Console:<br><br>1. Choose **Business Processes** > **Manager**.<br><br>2. Find the FTPClientDemoAllServices business process.<br><br>3. Run the FTPClientDemoAllServices business process with the following settings:<br>• Run As User = Admin<br>• Server filename = <install_directory>/installed_data/ psftpclient/FTPClientDemoImport.xml.<br><br>4. Verify that the business process runs successfullyDebug information for this service can be found in the FTP Client adapter and services log files. |
| Notes | Every FTP Client service returns a response code from the server. If this code is an error code as defined by the FTP specification (that is, 4xx or 5xx) then the business process will produce a fault. If the error code is expected, use an OnFault service to continue interacting with the trading partner. There are two exceptions to this rule:<br>• FTP Client GET service: If using the remoteFilePattern parameter and one of the files returns an error code indicating that the file could not be found, the GET command will continue without producing a fault. The error code will still be visible in the Transcript Document.<br>• FTP Client QUOTE service: This service never produces a fault, because the service does not know what constitutes a valid response from the quoted command. |

## Implementing the FTP Client SITE Service

To implement the FTP Client SITE service, complete the following tasks:

1. Create an FTP Client SITE service configuration (or enable the configuration installed with Sterling Integrator and edit parameters as needed). For information, see *Managing Services and Adapters*.
2. Configure the FTP Client SITE service. For information, see *Configuring the FTP Client SITE Service*.
3. Use the FTP Client SITE service in a business process.

## Configuring the FTP Client SITE Service

To configure the FTP Client SITE service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client SITE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SiteCommand | A site-specific control command. Required. Valid value is any site-specific command that does not require the use of a data connection. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Output from Service to Business Process

The following table contains the parameters passed from the FTP Client SITE service to the business process:

| Parameter | Description |
|---|---|
| ServerResponse | Indicates the FTP server response, which may include a reply code and any text associated with the reply code. Required. |
| TranscriptDocumentId | Identifies the document that contains a transcript of the exact exchange with the FTP server. Required. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the FTP Client SITE service:

| Parameter | Description |
|---|---|
| ResponseTimeout | Maximum number of seconds the FTP client waits for the server to respond before the session times out and terminates. Optional. Valid value is any numeric value. The default is the value from the FTP Client Begin Session service ResponseTimeout parameter. Minimum value you can specify is 1 second. If the value you specify is less than 1 second, the FTP Client SITE service resets the value to 1 second. |
| SessionToken | Specifies the identifier for the session established between the FTP Client adapter and an FTP server. Required.<br>**Note:** The session token is returned from the FTP Client Begin Session service. |
| SiteCommand | A site-specific control command. Required. Valid value is any site-specific command that does not require the use of a data connection. |
| SaveTranscript | Indicates how to handle the transcript. Valid values are:<br>• erroronly – persists the transcript only when an error occurs<br>• on – always persists the transcript<br>Default is on. Optional. |

## Business Process Example

The following example business process illustrates using the FTP Client SITE service:

```
<sequence>
  [[ Insert FTP Client Begin Session ]]
    <operation name="FTP SITE SERVICE">
    <participant name="FTPClientSite"/>
    <output message="SiteRequest">
       <assign to="SessionToken"
        from="/ProcessData/FtpBeginSessionServiceResults/SessionToken/text()">
       </assign>
       <assign to="SiteCommand">Help</assign>
    </output>
    <input message="inmsg">
       <assign to="FTPClientSiteResults" from="*"></assign>
    </input>
    </operation>
  [[ Insert FTP Client End Session ]]
</sequence>
```

# Chapter 89. FTP Reverse Proxy Adapter

The FTP Reverse Proxy adapter provides a high level of data protection between external connections using FTP and your company's Sterling Integrator server. Use this adapter to receive documents from a trading partner who uses the FTP protocol.

The FTP Reverse Proxy adapter can only be used with the Sterling Secure Proxy 2.0 (or later) product. See the Sterling Secure Proxy documentation for information.

# Chapter 90. FTP Server Adapter (V5100 - 5101)

The FTP Server adapter receives and processes requests from external trading partners that are submitted using the FTP protocol. This adapter is used with a Perimeter server. The following table provides an overview of the FTP Server adapter:

| System name | FTP Server Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | None |
| Description | This adapter receives and processes requests from external trading partners that are submitted using the FTP protocol. This adapter is used with a Perimeter server. |
| Business usage | Use this adapter to get or put files from:<br>• Mailbox in this system<br>• Physical file system<br><br>No additional permissions are required. |
| Usage example | A trading partner uses an FTP client to retrieve a business document from a mailbox. The FTP Server adapter receives and processes the trading partner request. |
| Preconfigured? | A configuration of the FTP Server adapter is installed, but is disabled by default. You can enable the preconfigured FTP Server adapter or create a new configuration. |
| Requires third party files? | Certicom SSL Library (currently available in the system) |
| Platform availability | All supported platforms |
| Related services | None |
| Application requirements | To log in to the FTP server, you must have permission to your virtual root (either explicitly assigned or defaulted).<br><br>To access a mailbox, you must have permission to that mailbox and all mailboxes that may be between it and your virtual root. |
| Initiates business processes? | The FTP Server adapter:<br>• 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.<br>• Does not initiate business processes if the Payload Repository is a mailbox. However, mailbox activities can trigger routing rules. |
| Invocation | Not used in business processes |
| Business process context considerations | None |
| Returned status values | None |

| Restrictions | Restrictions: |
|---|---|
| | • FTP Server is tightly integrated with the systems's mailbox system. An FTP client can only access the mailbox that is assigned to its user account. |
| | • FTP Server does not support all functions specified in RFC 0959 (Standard FTP Server). It supports basic functions to integrate with the system mailbox system such as list message and sub-mailbox, send and extract message to/from mailbox. |
| | • FTP Server is not integrated with business process invocation when processing a request from a client. |
| | • The home directory for FTP is a virtual root mailbox in the system. Mailboxes include both extractable and nonextractable messages. When accessing a mailbox using the FTP Server adapter, only extractable messages are displayed. To change this default behavior, edit the ftpserver.properties file and set `listUnextractables=true` (Default is false). |
| | • The timeout value for a control channel connection is controlled by a parameter in the ftpserver.properties file. The default timeout value is 600 seconds. The minimum value is 60 seconds. If the control channel is idle longer than the timeout value, the session is terminated, unless the data channel is open (whether or not data is being transferred). |
| | • To access the FTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root. |
| | • Restricted operation can be granted to users with a parameter named **MailboxLoginWithoutVirtualRootPermission**. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes. |
| Restrictions | When the FTP Server adapter is used with File System, the following restrictions apply: |
| | • The FTP Server is tightly integrated with the system's file system. An FTP client can only access the directory that is assigned to its user account. |
| | • The FTP Server supports almost all functions specified in RFC 0959 (Standard FTP Server). |
| | • The timeout value for a control channel connection is controlled by a parameter in the ftpserver.properties file. The default timeout value is 600 seconds. The minimum value is 60 seconds. If the control channel is idle longer than the timeout value, the session is terminated, unless the data channel is open (whether or not data is being transferred). |
| | • The home directory for FTP user is a combination of Base Directory and File System Virtual Root specified for the user. Only the directories and files that are within this directory are accessible to the user. |
| | • When a particular user connects to different instances of FTP server adapter on the same instance, the user would see same files. This may not be true for FTP servers configured with File System. When the same user connects to different instances of FTP server on the same instance, the user would either see same files or different, depending on how the FTP server is configured. |

| Persistence level | None. This adapter does not have a pre-set persistence level. |
|---|---|
| Testing considerations | At startup, attempt to access the FTP server using a supported FTP client with the configured IP address and port. |
| | Debug information can be found in the FTP logs. Select Logging Level from the following: |
| | • Error – Errors only |
| | • Communication Trace – Errors, requests from clients, and responses from the Server adapter, including ACL violations |
| | • All - for debugging, all activities. |

## Implementing the FTP Server Adapter

To implement the FTP Server adapter, complete the following tasks:

1. Create an FTP Server adapter configuration (or enable the configuration installed with the application and edit parameters as needed).
2. Configure the FTP Server adapter.

## Configuring the FTP Server Adapter

To configure the FTP Server adapter, you must specify settings for the following fields:

| UI Field | Description |
|---|---|
| Name | Unique and meaningful name for the adapter configuration. Required. |
| Description | Meaningful description for the adapter configuration, for reference purposes. Required. |
| Select a Group | Not applicable for this adapter. Leave at default. |
| FTP Server Listen Port | The port number that the FTP Server should bind to and listen on for connection requests. The default value depends on the system platform and on your application configuration. Required. |
| Active Data Port Range | A range of ports that the server can allocate for the transfer of data to or from the FTP client in active mode. Optional. If left blank, the server will choose available system ports. Example values are: |
| | • 1024-2048 |
| | • 2222 |
| | • 3000-4000 |
| | • 10500-10599,10700-10799 |
| Passive Data Port Range | A range of ports that the server can allocate for the transfer of data to or from the FTP client in passive mode. Optional. If left blank, the server will choose available system ports. Example values are: |
| | • 1024-2048 |
| | • 2222 |
| | • 3000-4000 |
| | • 10500-10599,10700-10799 |

| UI Field | Description |
|---|---|
| Perimeter Server | Select a Perimeter server from the list. Default is node1 & local. Required. **Note:** You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring. |
| Transfer Buffer Size (bytes) | Specifies the size in bytes of the buffer used when transferring a file. Required. Valid values are 0 to 9,999,999,999. Default is 32000. |
| Minimum Number of Threads | A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be less than or equal to the Maximum Number of Threads value. Default is 3. Required. **Note:** Retain the default value unless instructed otherwise by Sterling Commerce support. |
| Maximum Number of Threads | A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be greater than or equal to the Minimum Number of Threads value. Default is 6. Required. **Note:** Retain the default value unless instructed otherwise by Sterling Commerce support. |
| Resumption Timeout (hours) | Timeout value for the incomplete document before it is purged. Required. Valid value is any number between 1 and 9,999,999. |
| NAT Address | Specifies the NAT IP address that the FTP server should send to the user FTP client in the passive connection mode. Optional. Overrides the global NAT address specified in the ftpserver.properties file. |
| Maximum Logins | Maximum number of logins the adapter may have active at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999. |
| Maximum Logins per User | Maximum number of logins each user may have active on this adapter at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999. |
| Payload Repository | Whether files or messages will be stored in a mailbox or a physical file system on the server. Required. Valid values are:<br><br>• Mailbox (default) - If you want to restrict user access to specific mailboxes, see the *Mailbox Features, Creating Virtual Roots* documentation.<br><br>• File System - If you want to restrict user access to specific file system folders and subfolders, see the *Configuring an File System Virtual Root*. |
| Document Storage | Displayed only if Mailbox is selected for Payload Repository. Indicates whether the body of the requested document must be stored on the file system or if it should be in the database. Required. Valid values are:<br><br>• System Default – If your system administrator has changed the installed default of File System, this ensures that the correct location is used.<br><br>• Database – Body of the request document will be stored in the database.<br><br>• File System (default) – This is the default value when the application is installed, but it can be changed. Contact your system administrator to see if the default has been changed.<br><br>**Note:** For more information about document storage types, see *Managing Services and Adapters*. |
| Add Policy Type | If you want to apply an existing policy to this instance, select the plus sign. |

| UI Field | Description |
|---|---|
| Select policy type | Select one of the adapter policy types:<br>• Bandwidth Limiting Policy<br>• Lockout Policy<br>• Data Limit Policy<br>• Command Limiting Policy |
| Select Policy | Select from the list. Policy must have already been created. |
| Select Business Process Base Directory | Parameter is only configurable if File System is selected for Payload Repository. Choose the business process from the list to be invoked each time an inbound file is received. Optional. |
| Base Directory | Parameter is only configurable if File System is selected for Payload Repository. Path to the directory on the physical file system which this server adapter has access to. The file system virtual root defined for any user should be relative to this directory. The home directory for any user will be a combination of this directory and the file system virtual root. Required. The operating system level user who is running the JVM must have access to this directory. |
| 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 FTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages. |
| Should the restricted users be assigned a specific range of ports? | Select Yes or No to indicate whether to assign a specific port, range, or ranges of ports to the users. Required. Default is No. If Yes, specify *User* Active Ports, *User* Passive Ports, *Group* Active Ports, and or *Group* Passive Ports on subsequent pages. You can specify any or all of these fields. |
| Should users start in the directory that matches their user name upon login? | Valid values are:<br>• 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.<br>• 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. |
| *User* Active Ports | Any port number, range, or ranges of port numbers to be used as ACTIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |
| *User* Passive Ports | Any port number, range, or ranges of port numbers to be used as PASSIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |

| UI Field | Description |
|---|---|
| *Group* Active Ports | Any port number, range, or ranges of port numbers to be used as ACTIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |
| *Group* Passive Ports | Any port number, range, or ranges of port numbers to be used as PASSIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |
| Extractable Count | The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer. Optional. |
| Extractable For | Indicates the length of time (in days, hours and minutes) the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable Count. Valid value is in the format *dddhhmm*. Optional. |
| Extractable | Whether the message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Optional. Valid values are:<br>• Yes (Default)<br>• No |
| SSL | Whether Secure Sockets Layer (SSL) is active. Required. Valid values are:<br>• None – If SSL is requested by a client it will be rejected. (Default)<br>• Optional – SSL is used if requested by a client.<br>• Must – Clients that do not request SSL are not allowed to authenticate.<br>**Note:** If Optional or Must is specified, the asset protection key must enable SSL for the appropriate protocol. |
| Key Certificate Passphrase | Password that protects the server key certificate. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional. |
| Cipher Strength | Strength of the algorithms used to encrypt data. Required if SSL option is Must or Optional. Valid values are:<br>• ALL<br>• WEAK – Often required for international e-commerce, because government regulations prohibit STRONG encryption from being exported.<br>• STRONG – Default. |
| Key Certificate (System Store) | Private key and certificate for server authentication. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional. |
| CA Certificates | Certificate used to validate the certificate of an FTP client. This is the public key. If no CA certificate is chosen, no client certification is performed. Optional. |
| Clear Command Channel | Indicates that communication across the command channel is not encrypted after authentication is completed. Optional. |

## Applying Policies to the FTP Adapter

You can apply adapter polices to the FTP 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, see *Adapter Policies*.

## FTP Server Functions Supported

The following table lists the FTP functions that are supported with the FTP Server adapter:

| Category | Commands Supported |
|---|---|
| Access Control commands | • USER – User name<br>• PASS – Password<br>• CWD – Change Working Directory<br>• CDUP – Change to Parent Directory<br>• QUIT – Logout<br>• REIN – Reinitialize |
| Transfer Parameter Commands | • PORT – Data port<br>• PASV – Passive mode<br>• TYPE – Representation type (ASCII, Binary, EBCDIC, and Local-8)<br>• STRU – File Structure (File)<br>• MODE – Transfer mode (Streamed and Deflated) |
| Service Commands | • DELE – Delete<br>• RETR – Retrieve<br>• STOR – Store<br>• ABOR – Abort<br>• PWD – Print Working Directory<br>• XPWD – Print Working Directory (legacy format)<br>• LIST – List<br>• NLST – Name List<br>• HELP – Help<br>• NOOP – No Operation<br>• RNFR – Rename From<br>• RNTO – Rename To<br>• SYST – System<br>• MDTM – Last-modified time of a given file on a remote host<br>• STOU – Store Unique<br>• APPE – Append<br>• ALLO – Allocate<br>• REST – Restart<br>• RMD – Remove Directory<br>• XRMD – Remove Directory (legacy format)<br>• MKD – Make Directory<br>• XMKD – Make Directory (legacy format)<br>• SITE – Site Parameter (CPWD, HELP, PSWD, WHO, ZONE)<br>• STAT – Status |

| Category | Commands Supported |
|---|---|
| Security Commands | • AUTH – Authentication/Security Mechanism<br>• CCC – Clear Command Channel<br>• PBSZ – Protect Buffer Size<br>• PROT – Data Channel Protection Level<br>• EPRT - Specifies an extended address and port to which the server should connect<br>• EPSV - Enter extended passive mode<br>• SIZE - Return the size of a file |

## FTP Server Functions Not Supported

The following table lists the FTP functions that are not supported with the FTP Server adapter:

| Category | Commands Not Supported |
|---|---|
| Access Control commands | • ACCT – Account<br>• SMNT – Structure Mount |
| Transfer Parameter Commands | • TYPE – Representation type (Local Byte other than L8)<br>• STRU – File Structure (Record and Page)<br>• MODE – Transfer mode (Block and Compressed ) |

## Activity Types for the FTP Server Adapter

This adapter reports the following activities to the Services Controller for activity monitoring:
- PUT – Adds a file to a mailbox
- GET – Retrieves a file from a mailbox
- Session – Records all activity after connection

## File System Virtual Root

When you configure an FTP 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 file system 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**.

# Chapter 91. FTP Server Adapter (V5102 and higher)

The FTP Server adapter receives and processes requests from external trading partners that are submitted using the FTP protocol. This adapter is used with a Perimeter server. The following table provides an overview of the FTP Server adapter:

| System name | FTP Server Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | None |
| Description | This adapter receives and processes requests from external trading partners that are submitted using the FTP protocol. This adapter is used with a Perimeter server. |
| Business usage | Use this adapter to get or put files from:<br>• Mailbox in this system<br>• Physical file system<br><br>No additional permissions are required. |
| Usage example | A trading partner uses an FTP client to retrieve a business document from a mailbox. The FTP Server adapter receives and processes the trading partner request. |
| Preconfigured? | A configuration of the FTP Server adapter is installed, but is disabled by default. You can enable the preconfigured FTP Server adapter or create a new configuration. |
| Requires third party files? | Certicom SSL Library (currently available in the system) |
| Platform availability | All supported platforms |
| Related services | None |
| Application requirements | To log in to the FTP server, you must have permission to your virtual root (either explicitly assigned or defaulted).<br><br>To access a mailbox, you must have permission to that mailbox and all mailboxes that may be between it and your virtual root. |
| Initiates business processes? | The FTP Server adapter:<br>• 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.<br>• Does not initiate business processes if the Payload Repository is a mailbox. However, mailbox activities can trigger routing rules. |
| Invocation | Not used in business processes |
| Business process context considerations | None |
| Returned status values | None |

| Restrictions | Restrictions: |
|---|---|
| | • FTP Server is tightly integrated with the systems's mailbox system. An FTP client can only access the mailbox that is assigned to its user account. |
| | • FTP Server does not support all functions specified in RFC 0959 (Standard FTP Server). It supports basic functions to integrate with the system mailbox system such as list message and sub-mailbox, send and extract message to/from mailbox. |
| | • FTP Server is not integrated with business process invocation when processing a request from a client. |
| | • The home directory for FTP is a virtual root mailbox in the system. Mailboxes include both extractable and nonextractable messages. When accessing a mailbox using the FTP Server adapter, only extractable messages are displayed. To change this default behavior, edit the ftpserver.properties file and set `listUnextractables=true` (Default is false). |
| | • The timeout value for a control channel connection is controlled by a parameter in the ftpserver.properties file. The default timeout value is 600 seconds. The minimum value is 60 seconds. If the control channel is idle longer than the timeout value, the session is terminated, unless the data channel is open (whether or not data is being transferred). |
| | • To access the FTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root. |
| | • Restricted operation can be granted to users with a parameter named **MailboxLoginWithoutVirtualRootPermission**. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes. |
| Restrictions | When the FTP Server adapter is used with File System, the following restrictions apply: |
| | • The FTP Server is tightly integrated with the system's file system. An FTP client can only access the directory that is assigned to its user account. |
| | • The FTP Server supports almost all functions specified in RFC 0959 (Standard FTP Server). |
| | • The timeout value for a control channel connection is controlled by a parameter in the ftpserver.properties file. The default timeout value is 600 seconds. The minimum value is 60 seconds. If the control channel is idle longer than the timeout value, the session is terminated, unless the data channel is open (whether or not data is being transferred). |
| | • The home directory for FTP user is a combination of Base Directory and File System Virtual Root specified for the user. Only the directories and files that are within this directory are accessible to the user. |
| | • When a particular user connects to different instances of FTP server adapter on the same instance, the user would see same files. This may not be true for FTP servers configured with File System. When the same user connects to different instances of FTP server on the same instance, the user would either see same files or different, depending on how the FTP server is configured. |

| Persistence level | None. This adapter does not have a pre-set persistence level. |
|---|---|
| Testing considerations | At startup, attempt to access the FTP server using a supported FTP client with the configured IP address and port.<br><br>Debug information can be found in the FTP logs. Select Logging Level from the following:<br><br>• Error – Errors only<br>• Communication Trace – Errors, requests from clients, and responses from the Server adapter, including ACL violations<br>• All - for debugging, all activities. |

## Implementing the FTP Server Adapter

To implement the FTP Server adapter, complete the following tasks:

1. Create an FTP Server adapter configuration (or enable the configuration installed with the application and edit parameters as needed).
2. Configure the FTP Server adapter.

## Configuring the FTP Server Adapter

To configure the FTP Server adapter, you must specify settings for the following fields:

| UI Field | Description |
|---|---|
| Name | Unique and meaningful name for the adapter configuration. Required. |
| Description | Meaningful description for the adapter configuration, for reference purposes. Required. |
| Select a Group | Not applicable for this adapter. Leave at default. |
| FTP Server Listen Port | The port number that the FTP Server should bind to and listen on for connection requests. The default value depends on the system platform and on your application configuration. Required. |
| Active Data Port Range | A range of ports that the server can allocate for the transfer of data to or from the FTP client in active mode. Optional. If left blank, the server will choose available system ports. Example values are:<br><br>• 1024-2048<br>• 2222<br>• 3000-4000<br>• 10500-10599,10700-10799 |
| Passive Data Port Range | A range of ports that the server can allocate for the transfer of data to or from the FTP client in passive mode. Optional. If left blank, the server will choose available system ports. Example values are:<br><br>• 1024-2048<br>• 2222<br>• 3000-4000<br>• 10500-10599,10700-10799 |

| UI Field | Description |
|---|---|
| Perimeter Server | Select a Perimeter server from the list. Default is node1 & local. Required. **Note:** You should use a specific external interface for communications with trading partners. Using a wildcard address can cause problems with FTP sessions. If some other process has bound the port used for the data channel on an interface, it may receive connections intended for the data channel. Using a specific TCP/IP address or DNS name prevents this from occurring. |
| Transfer Buffer Size (bytes) | Specifies the size in bytes of the buffer used when transferring a file. Required. Valid values are 0 to 9,999,999,999. Default is 32000. |
| Minimum Number of Threads | A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be less than or equal to the Maximum Number of Threads value. Default is 3. Required. **Note:** Retain the default value unless instructed otherwise by Sterling Commerce support. |
| Maximum Number of Threads | A tuning parameter that indicates the range of threads available for handling events to improve performance. Must be greater than or equal to the Minimum Number of Threads value. Default is 6. Required. **Note:** Retain the default value unless instructed otherwise by Sterling Commerce support. |
| Resumption Timeout (hours) | Timeout value for the incomplete document before it is purged. Required. Valid value is any number between 1 and 9,999,999. |
| NAT Address | Specifies the NAT IP address that the FTP server should send to the user FTP client in the passive connection mode. Optional. Overrides the global NAT address specified in the ftpserver.properties file. |
| Maximum Logins | Maximum number of logins the adapter may have active at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999. |
| Maximum Logins per User | Maximum number of logins each user may have active on this adapter at any point of time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999. |
| Payload Repository | Whether files or messages will be stored in a mailbox or a physical file system on the server. Required. Valid values are:<br>• Mailbox (default) - If you want to restrict user access to specific mailboxes, see the *Mailbox Features, Creating Virtual Roots* documentation.<br>• File System - If you want to restrict user access to specific file system folders and subfolders, see the *Configuring an File System Virtual Root*. |
| Document Storage | Displayed only if Mailbox is selected for Payload Repository. Indicates whether the body of the requested document must be stored on the file system or if it should be in the database. Required. Valid values are:<br>• System Default – If your system administrator has changed the installed default of File System, this ensures that the correct location is used.<br>• Database – Body of the request document will be stored in the database.<br>• File System (default) – This is the default value when the application is installed, but it can be changed. Contact your system administrator to see if the default has been changed.<br>**Note:** For more information about document storage types, see *Managing Services and Adapters*. |
| Add Policy Type | If you want to apply an existing policy to this instance, select the plus sign. |

| UI Field | Description |
|---|---|
| Select policy type | Select one of the adapter policy types:<br>• Bandwidth Limiting Policy<br>• Lockout Policy<br>• Data Limit Policy<br>• Command Limiting Policy |
| Select Policy | Select from the list. Policy must have already been created. |
| Select Business Process Base Directory | Parameter is only configurable if File System is selected for Payload Repository. Choose the business process from the list to be invoked each time an inbound file is received. Optional. |
| Base Directory | Parameter is only configurable if File System is selected for Payload Repository. Path to the directory on the physical file system which this server adapter has access to. The file system virtual root defined for any user should be relative to this directory. The home directory for any user will be a combination of this directory and the file system virtual root. Required. The operating system level user who is running the JVM must have access to this directory. |
| 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 FTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages. |
| Should the restricted users be assigned a specific range of ports? | Select Yes or No to indicate whether to assign a specific port, range, or ranges of ports to the users. Required. Default is No. If Yes, specify *User* Active Ports, *User* Passive Ports, *Group* Active Ports, and or *Group* Passive Ports on subsequent pages. You can specify any or all of these fields. |
| Should users start in the directory that matches their user name upon login? | Valid values are:<br>• 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.<br>• 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. |
| *User* Active Ports | Any port number, range, or ranges of port numbers to be used as ACTIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |
| *User* Passive Ports | Any port number, range, or ranges of port numbers to be used as PASSIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |

| UI Field | Description |
|---|---|
| *Group* Active Ports | Any port number, range, or ranges of port numbers to be used as ACTIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |
| *Group* Passive Ports | Any port number, range, or ranges of port numbers to be used as PASSIVE port. Optional. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are:<br>• 3000<br>• 4000-5000, 6000 |
| Extractable Count | The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Valid value is any integer. Optional. |
| Extractable For | Indicates the length of time (in days, hours and minutes) the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable Count. Valid value is in the format *dddhhmm*. Optional. |
| Extractable | Whether the message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Optional. Valid values are:<br>• Yes (Default)<br>• No |
| SSL | Whether Secure Sockets Layer (SSL) is active. Required. Valid values are:<br>• None – If SSL is requested by a client it will be rejected. (Default)<br>• Optional – SSL is used if requested by a client.<br>• Must – Clients that do not request SSL are not allowed to authenticate.<br>**Note:** If Optional or Must is specified, the asset protection key must enable SSL for the appropriate protocol. |
| Key Certificate Passphrase | Password that protects the server key certificate. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional. |
| Cipher Strength | Strength of the algorithms used to encrypt data. Required if SSL option is Must or Optional. Valid values are:<br>• ALL<br>• WEAK – Often required for international e-commerce, because government regulations prohibit STRONG encryption from being exported.<br>• STRONG – Default. |
| Key Certificate (System Store) | Private key and certificate for server authentication. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional. |
| CA Certificates | Certificate used to validate the certificate of an FTP client. This is the public key. If no CA certificate is chosen, no client certification is performed. Optional. |
| Clear Command Channel | Indicates that communication across the command channel is not encrypted after authentication is completed. Optional. |

| UI Field | Description |
|---|---|
| Support for concurrent duplicate named file transfers | Allows sending duplicate-named files to the same Mailbox using the same username. It also allows partners to receive multiple duplicate files with the same name, concurrently. Valid values are:<br><br>• Limited (resume of file transfers) – The file transfer can be resumed if transfer fails from the point of failure. You cannot transfer files with the same name concurrently using the same Mailbox and the same username. Default.<br><br>• Full, concatenate duplicate-named files on a GET (resume of file transfers not supported) – Supports sending files with the same name concurrently using the same Mailbox and the same username. The files with the same name are concatenated on a GET operation. Listing shows a single concatenated file. You cannot resume broken file transfers.<br><br>• Full (resume of file transfers not supported) – Supports sending files with the same name concurrently using the same Mailbox. The files with the same name are not concatenated on both GET or PUT operations. Listing shows multiple files at the client side. You cannot resume broken file transfers. |

## Applying Policies to the FTP Adapter

You can apply adapter polices to the FTP 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, see *Adapter Policies*.

## FTP Server Functions Supported

The following table lists the FTP functions that are supported with the FTP Server adapter:

| Category | Commands Supported |
|---|---|
| Access Control commands | • USER – User name<br>• PASS – Password<br>• CWD – Change Working Directory<br>• CDUP – Change to Parent Directory<br>• QUIT – Logout<br>• REIN – Reinitialize |
| Transfer Parameter Commands | • PORT – Data port<br>• PASV – Passive mode<br>• TYPE – Representation type (ASCII, Binary, EBCDIC, and Local-8)<br>• STRU – File Structure (File)<br>• MODE – Transfer mode (Streamed and Deflated) |

| Category | Commands Supported |
|---|---|
| Service Commands | • DELE – Delete<br>• RETR – Retrieve<br>• STOR – Store<br>• ABOR – Abort<br>• PWD – Print Working Directory<br>• XPWD – Print Working Directory (legacy format)<br>• LIST – List<br>• NLST – Name List<br>• HELP – Help<br>• NOOP – No Operation<br>• RNFR – Rename From<br>• RNTO – Rename To<br>• SYST – System<br>• MDTM – Last-modified time of a given file on a remote host<br>• STOU – Store Unique<br>• APPE – Append<br>• ALLO – Allocate<br>• REST – Restart<br>• RMD – Remove Directory<br>• XRMD – Remove Directory (legacy format)<br>• MKD – Make Directory<br>• XMKD – Make Directory (legacy format)<br>• SITE – Site Parameter (CPWD, HELP, PSWD, WHO, ZONE)<br>• STAT – Status |
| Security Commands | • AUTH – Authentication/Security Mechanism<br>• CCC – Clear Command Channel<br>• PBSZ – Protect Buffer Size<br>• PROT – Data Channel Protection Level<br>• EPRT - Specifies an extended address and port to which the server should connect<br>• EPSV - Enter extended passive mode<br>• SIZE - Return the size of a file |

## FTP Server Functions Not Supported

The following table lists the FTP functions that are not supported with the FTP Server adapter:

| Category | Commands Not Supported |
|---|---|
| Access Control commands | • ACCT – Account<br>• SMNT – Structure Mount |
| Transfer Parameter Commands | • TYPE – Representation type (Local Byte other than L8)<br>• STRU – File Structure (Record and Page)<br>• MODE – Transfer mode (Block and Compressed ) |

## Activity Types for the FTP Server Adapter

This adapter reports the following activities to the Services Controller for activity monitoring:

- PUT – Adds a file to a mailbox
- GET – Retrieves a file from a mailbox
- Session – Records all activity after connection

## File System Virtual Root

When you configure an FTP 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 file system 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**.

# Chapter 92. Get Document Info Service

The Get Document Info service provides information about the primary document. You can query the service about many types of information, including DocumentName, DocumentBodyLength, and DocumentID. The following table provides an overview of the Get Document Info service:

| System name | GetDocumentInfoServiceType |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | The Get Document Info service provides information about the primary document. You can query the service about the following information:<br><br>• DocumentName<br>• DocumentBodyLength<br>• DocumentId<br>• DocumentSubject<br>• DocumentCreateTime<br>• DocumentContentType<br>• DocumentCharEncoding<br>• DocumentContentSubType<br>• DocumentPreviousDocId<br>• DocumentLength<br>• Correlations<br>• Storage Type<br>• Node name<br>• File name<br>• Storage time<br><br>You can also use the service to set the following information, with or without creating a new document in the system:<br><br>• BodyName<br>• Document Content Type<br>• DocumentCharEncoding<br>• DocumentContentSubType<br>• updateMetaDataOnly: When set to "true" BodyName, Document Content Type, DocumentCharEncoding, DocumentContentSub Type, Document Subject, DocumentName, and DocumentSize can be updated without creating a new document. |
| Business Usage | This service allows information obtained from the document to be used in the business process at run time. You can also use the service to set certain information about the document in the business process. |
| Usage Example | Obtain correlations from a document, such as a correlation for document type, and use this to determine further processing that would be specific to that document type. |
| Preconfigured? | Yes |

| | |
|---|---|
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | None |
| Application requirements | None |
| Initiates Business Processes | This service does not initiate a business process |
| Invocation | Not applicable |
| Business Process Considerations | Allows a document ID to be passed instead of the whole document. This makes information from the document available for use in the process without opening the document. |
| Returned Status Values | Returned status values:<br>• Success<br>• Error |
| Restrictions | None |
| Persistence Level | System default |

## Implementing the Get Document Info Service

To implement the Get Document Info service, complete the following tasks:

1. Create a Get Document Info service configuration. See *Creating a Service Configuration*.
2. Configure the Get Document Info service. See *Configuring the Get Document Info Service*.
3. Use the Get Document Info service in a business process.

## Configuring the Get Document Info Service

To configure the Get Document Info service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| BodyName | Body Name of the document. Optional. |
| CORRELATION | Whether to pass correlations associated with the document. Optional. Valid values:<br>• true – Pass document correlation parameters<br>• false – Do not pass document correlation parameters<br>Default is false. |
| DOCUMENT_ID | Document ID to look up from the workflow context. If there is no document ID, it will look up the primary document instead. Optional. Valid value: *hostname*:e52fd2:fd70c53c8c:-7648.<br>**Note:** When you specify DOCUMENT_ID, the document specified is loaded into the Primary Document. If a Primary Document already exists, it will be overwritten. |
| DocumentCharEncoding | Type of character encoding used in the document. Optional. Example: iso-8859-1. |
| DocumentContentSubType | Content subtype of the document. Optional. Example: plain. |

| Field | Description |
|---|---|
| DocumentContentType | Content type of the document. Optional. Example: text. |
| Encrypted | Indicates if the document is encrypted. Optional. |

Once configured, these parameters are passed from the business process to the Get Document Info service.

## Parameters Passed from Business Process to Service

The following parameters are passed from the Business Process to the Get Document Info service:

| Field | Description |
|---|---|
| DocumentId | Document ID to look up from the workflow context. If there is no document ID, it will look up the primary document instead. Optional. Example: *hostname*:e52fd2:fd70c53c8c:-7648. |
| DocumentContentType | The current document content type. Optional. |
| DocumentContentSubType | The current document content sub-type. Optional. |
| DocumentCharEncoding | The current document character encoding. Optional |
| Correlation | The current document correlation. Optional. |
| BodyName | The current document body name. Optional |
| doClone | Set to true to modify the document metadata in the clone or copy of the existing document. The modified data is in the cloned document. Valid values are true and false. Optional. For example, when you alter the value in DocumentContentType, setting doClone to true creates a clone of the existing document but with the modified value. If you do not set the doClone parameter or set it to false, the existing document will have the new metadata value and the modified value is available in the process data. |

| Field | Description |
|---|---|
| updateMetaDataOnly | A flag that indicates if the document data is for a metadata update only. Optional. Valid values are: True or False. If set to true, the service can set the following metadata without creating a new document that is stored in the database:<br><br>• doc name<br>• bodyName<br>• char_encoding<br>• subject<br>• content_type<br>• content_subtype<br>• doc size<br><br>Default is false. Example of a document with updateMetaDataOnly set to true:<br><br>```<operation name="SetContentType">\n<participant name="GetDocumentInfoService"/>\n<output message="xout">\n<assign to="." from="*"/>\n<assign to="DocumentContentType">text</assign>\n<assign to="DocumentContentSubType">plain\n</assign>\n<assign to="DocumentCharEncoding">iso-8859-1\n</assign>\n<assign to="updateMetaDataOnly">true</assign>\n</output>\n<input message="xin">\n<assign to="." from="*"/>\n</input>\n</operation>```<br><br>Example of a document that is not using updateMetaDataOnly and is set to false, or not set at all. A new document will be saved in the database:<br><br>```<operation name="SetContentType">\n<participant name="GetDocumentInfoService"/>\n<output message="xout">\n<assign to="." from="*"/>\n<assign to="DocumentContentType">text\n</assign>\n<assign to="DocumentContentSubType">plain\n</assign>\n<assign to="DocumentCharEncoding">iso-8859-1\n</assign>\n<assign to="updateMetaDataOnly">false</assign>\n</output>\n<input message="xin">\n<assign to="." from="*"/>\n</input>\n</operation>``` |

## Parameters Passed from Service to Business Process

The following parameters are passed from the Get Document Info service to the business process:

| Field | Description |
|---|---|
| PrimaryDocument | The link of the primary document. Optional. Example: *hostname*:73ce5c:ff993119fc:-733f |
| BodyName | The current body name. Required. |
| BodyLength | The current body length. Required. Default is 0. |
| DocumentBodyLength | The current document body length. Required. Default is 0. |
| DocumentName | The current document name. Optional. |
| DocumentId | The current document ID. Required. Example: *hostname*:331059:ff998a9313:-7f71 |
| DocumentSubject | The current document subject. Required. |
| DocumentCreateTime | The current document creation time. Required. |
| DocumentContentType | The current document content type. Optional. |
| DocumentCharEncoding | The current document character encoding. Optional. |
| DocumentContentSubType | The current document content sub-type. Optional. |
| DocumentPreviousDocId | The previous document ID. Optional. |
| DocumentLength | The current document length. Optional. |
| DocumentLifeSpan | The current document life span. Optional. |
| DocumentInitialId | The current document initial ID. Optional. |
| DocumentMaxInlineBodySize | The current document maximum inline body size. Optional. |
| DocumentPurgeAfter | The current document purge after. Optional. Valid values: any timestamp value. |
| DocumentStorageType | The current document storage type. Optional. |
| DocumentTrackingId | The current document tracking ID. Optional. |
| DocumentWorkflowId | The current document workflow ID. Optional. |
| DocumentFileName | The current document file name. Optional. |
| DocumentNodeName | The current document node name. Optional. |
| DocumentType | The current document type. Optional. |
| Encrypted | Indicates if the document is encrypted. Optional. |
| UpdateMetaDataOnly | Indicates if the document data is for a metadata update only. Optional. |

## Business Process Examples

### Example 1

Example of the getdocinfo service that shows processing of the primary document created by the xmlencoder.

```
<process name="GetDocumentInfoService_01">
    <sequence name="simple">
    <operation name="Set Document">
    <participant name="XMLEncoder"/>
    <output message="XMLEncoderTypeInputMessage">
      <assign to="mode">process_data_to_document</assign>
      <assign to="root_element">Document1</assign>
      <assign to="xPath">/ProcessData</assign>
      <assign to="." from="*"/>
```

```
      </output>
      <input message="inmsg">
        <assign to="." from="*"/>
      </input>
      </operation>
      <operation name="SetContentType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="." from="*"/>
        <assign to="DocumentContentType">text</assign>
        <assign to="DocumentContentSubType">plain</assign>
        <assign to="DocumentCharEncoding">iso-8859-1</assign>
      </output>
      <input message="xin">
        <assign to="." from="*"/>
      </input>
    </operation>
    </sequence>
</process>
```

## Example 2

Example of a process document that is collected by the filesystemadapter. The
second getinfoservice is processing a document by a document ID.

```
<process name="GetDocumentInfoService_02">
  <sequence name="simple">
  <!-- Create File System adapter instances for input and output -->
  <!-- Place the document you want processed in the input instance -->
    <operation name="FileSystemAdapter">
    <participant name="FileSystemAdapter"/>
    <output message="outputMessage">
      <assign to="Action">FS_COLLECT</assign>
      <assign to ="filter">My.log</assign>
      <assign to ="useSubFolders">false</assign>
      <assign to ="docStorageType">fs</assign>
      <assign to ="bootstrap">false</assign>
      <assign to ="deleteAfterCollect">false</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inputMessage">
      <assign to="." from="*"></assign>
  </input>
  </operation>
    <operation name="FileSystemAdapter">
    <participant name="FileSystemAdapter"/>
    <output message="outputMessage">
       <assign to="Action">FS_EXTRACT</assign>
       <assign to ="useSubFolders">false</assign>
       <assign to="." from="*"></assign>
  </output>
  <input message="inputMessage">
       <assign to="." from="*"></assign>
  </input>
  </operation>
    <operation name="GetDocumentInfoService">
    <participant name="GetDocumentInfoService"/>
    <output message="xout">
      <assign to="." from="*"/>
      <assign to="DocumentContentType">text</assign>
      <assign to="DocumentContentSubType">plain</assign>
      <assign to="DocumentCharEncoding">iso-8859-1</assign>
  </output>
  <input message="xin">
      <assign to="." from="*"/>
  </input>
  </operation>
```

```
              <operation name="GetDocumentInfoService">
              <participant name="GetDocumentInfoService"/>
              <output message="xout">
                <assign to="." from="*"/><
 <!-- The DOCUMENT_ID parameter must contain a correct value from -->
 <!-- the Document table in the business process database.-->
                  <assign to="DOCUMENT_ID">somehost:e61fd1:fd70c58c8c:-7478</assign>
                  <assign to="DocumentContentType">text</assign>
                  <assign to="DocumentContentSubType">plain</assign>
                  <assign to="DocumentCharEncoding">iso-8859-1</assign>
          </output>
          <input message="xin">
                  <assign to="." from="*"/>
          </input>
          </operation>
          </sequence>
</process>
```

## Example 3

Example of the correlation for the primary document that is created by the translator.

```
<process name = "GetDocumentInfoService_04">
<!-- GetDocumentInfoService_04 map must be checked in -->
<!-- This business process uses GetDocumentInfoService_04_input_data.xml -->
  <rule name="New">
    <condition>WF_CORRELATIONS/correlation/value/text() = 'N'</condition>
  </rule>
  <rule name="Not New">
    <condition>WF_CORRELATIONS/correlation/value/text() != 'N'</condition>
  </rule>
  <sequence>
    <sequence name="Translate and Report">
      <operation name="Translation">
         <participant name="Translation"/>
        <output message="TranslationTypeInputMessage">
          <assign to="map_name">PurchaseOrder_Metadata_Test</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
      <operation name="Correlation Service">
        <participant name="CorrelationService"/>
        <output message="CorrelationServiceTypeInputMessage">
          <assign to="NAME">BPMETA_MESSAGE_TYPE</assign>
          <assign to="VALUE">PurchaseOrder</assign>
          <assign to="TYPE">BUSINESS PROCESS</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
      <operation name="GetDocumentInfoService">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
          <assign to="CORRELATION">true</assign>
          <assign to="." from="*"/>
    </output>
    <input message="xin">
          <assign to="." from="*"/>
  </input>
      </operation>
    </sequence>
```

```
<choice name="Check POType">
  <select>
      <case ref="New" activity="ProcessNewPO"/>
      <case ref="Not New" activity="ProcessOtherPO"/>
  </select>
```

# Chapter 93. GS:Unix Purge Process Service

The following table provides an overview of the GS:Unix (Gentran:Server® for UNIX®) Purge Process service:

| System name | None |
|---|---|
| Graphical Process Modeler (GPM) category | None |
| Description | Deletes lifecycle records created before a date and time designated by you. |
| Preconfigured? | No |
| Requires third party files? | LC221 and LCDESTINFO tables created in Gentran:Server for UNIX. |
| Platform availability | Platform availability:<br>• Sun Solaris<br>• HP-UX<br>• IBM-AIX |
| Related services | No |
| Application requirements | Gentran:Server for UNIX |
| Initiates business processes? | No |
| Invocation | Runs by the Sterling Integrator scheduler. |
| Business process context considerations | No |

## How the GS:Unix Purge Process Service Works

The GS:Unix Purge Process service deletes records from the LIFECYCLE and LIFECYCLE_EXTENSION tables created in Sterling Integrator and from existing LC221 and LCDESTINFO tables created in Gentran:Server for UNIX. The service runs as a step in a business process or by itself on a schedule.

For example, consider the following scenario. Today is August 12, 2004 and the time is 3 p.m. You want to delete files older than three days and five hours. Using the GS:Unix Purge Process service, you can delete files created before August 9, 2004 at 10 a.m.

## Implementing the GS:Unix Purge Process Service

To implement the GS:Unix Purge Process service for use in a business process, complete the following tasks:

1. Activate your license for the GS:Unix Purge Process service. See *An Overview of Implementing Services*.
2. Create a GS:Unix Purge Process service configuration. See *Creating a Service Configuration*.
3. Configure the GS:Unix Purge Process service. See *Configuring the GS:Unix Purge Process Service*.
4. Use the GS:Unix Purge Process service in a business process.

## Configuring the GS:Unix Purge Process Service

To configure the GS:Unix Purge Process service, you must specify settings for the following fields in 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 service configuration. Required. |
| Description | Meaningful description for the service configuration, for reference purposes. Required. |
| Select a Group | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** See *Using Service Groups*. |
| Run As User | Applies to the scheduling of the business process. The Run As User field only displays as an option if **Start a business process once files are collected** is set to Yes. Type the user ID to associate with the schedule, or click the<br><br><br><br>icon and select a user ID from the list.<br><br>Valid values: Any valid Sterling Integrator user ID.<br>**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. |
| 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 | Information about scheduling the business process after the File System adapter collects files. The Schedule field only displays as an option if **Start a business process once files are collected** is set to Yes. Valid values:<br><br>• Do not use schedule If you select this field, the adapter does not start a business process and does not run on a schedule.<br><br>• Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup.<br><br>• Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup.<br><br>• Run based on days of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions.<br><br>• Run based on days of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |
| Lifespan Days (lifespandays) | Number of days from the current day used to determine which records to delete from the LIFECYCLE, LIFECYCLE_EXTENSION, LC221, and LCDESTINFO tables. Optional. |
| Lifespan Hours (lifespanhours) | Number of hours from the current time used to determine which records to delete from the LIFECYCLE, LIFECYCLE_EXTENSION, LC221, and LCDESTINFO tables. Optional. |
| Business Process (initialWorkFlowName) | Business process where the records are located. Optional. |

# Chapter 94. Gentran:Server Windows Adapter

The following table provides an overview of the Gentran:Server® Windows® adapter:

| System name | GenSrvNT |
|---|---|
| Graphical Process Modeler (GPM) category | All Services and Applications Sterling > Gentran |
| Description | Enables you to transfer data between Sterling Integrator and Gentran:Server for Windows as part of a business process within Sterling Integrator. |
| Preconfigured? | No |
| Requires third party files? | Requires third party files:<br>• An installed and configured copy of Gentran:Server for Windows version 3.1.1 or higher<br>• The host name of the computer where Gentran:Server for Windows is installed<br>• The name of the Gentran:Server for Windows mailbox that this adapter configuration will use to exchange data with Sterling Integrator<br>• The adapter client .jar file installed, configured, and running. |
| Platform availability | Platform availability:<br>• Microsoft Windows<br>• Sun Solaris<br>• HP-UX<br>• IBM-AIX<br>• United Linux<br>• RedHat AS |
| Related services | No |
| Application requirements | No |
| Initiates business processes? | Yes |
| Invocation | Runs by a scheduler or another adapter in a business process. |

## Requirements

To configure and use the Gentran:Server Windows adapter, you must be familiar with Gentran:Server for Windows and must have:

- An installed and configured copy of Gentran:Server for Windows version 3.1.1 or higher
- The host name of the computer where Gentran:Server for Windows is installed
- The name of the Gentran:Server for Windows mailbox that this adapter configuration will use to exchange data with Sterling Integrator
- The Gentran:Server Windows adapter client .jar file installed, configured, and running

## How the Gentran:Server Windows Adapter Works

The Gentran:Server Windows adapter is used in a business process to exchange data between Sterling Integrator and Gentran:Server for Windows.

The following sections summarize how the Gentran:Server Windows adapter works in a business process.

## Sterling Integrator Receives Data from Gentran:Server for Windows

The following procedure illustrates how Sterling Integrator receives data from Gentran:Server for Windows:

1. Using the scheduling parameters set for the adapter in Sterling Integrator, the Gentran:Server Windows adapter polls Gentran:Server for Windows to check if it has data in the Sterling Integrator mailbox.
2. If data is in the mailbox, the Gentran:Server Windows adapter retrieves the data through the remote RMI service of the adapter and the Sterling Integrator gateway.
3. When Sterling Integrator receives the data, the Gentran:Server Windows adapter starts the business process defined in the adapter configuration.

## Sterling Integrator Sends Data to Gentran:Server for Windows

The following procedure illustrates how Sterling Integrator sends data to Gentran:Server for Windows:

1. The Gentran:Server Windows adapter can be called in a business process to send data to Gentran:Server for Windows.
2. Using the Gentran:Server Windows adapter remote RMI service and Sterling Integrator, the data is sent to Gentran:Server for Windows.
3. The Sterling Integrator gateway creates the appropriate message and sends the data to the receiver.

## Implementing the Gentran:Server Windows Adapter

To implement the Gentran:Server Windows adapter, complete the following tasks:

1. Activate your license for the Gentran:Server Windows adapter. See *An Overview of Implementing Services*.
2. Install the Gentran:Server Windows adapter. See *Installing the Adapter Client on Gentran:Server for Windows*.
3. Install the adapter client on Gentran:Server for Windows. See *Installing the Adapter Client on Gentran:Server for Windows*.
4. Install Sterling Integrator Gateway on Gentran:Server for Windows. See *Installing Application Gateway in Gentran:Server for Windows*.
5. Create a Sterling Integrator Mailbox on Gentran:Server for Windows. See *Creating a Application Mailbox in Gentran:Server for Windows*.
6. Create a Gentran:Server Windows adapter configuration. See *Creating a Service Configuration*.
7. Configure the Gentran:Server Windows adapter. See *Configuring the Gentran:Server Windows Adapter*.
8. Use the configured services in a business process.

## Installing Java on Gentran:Server for Windows

Sterling Integrator installs a copy of the appropriate Java Development Kit (JDK) required for proper operation. You must ensure that there is a JDK installed on the machine where Gentran:Server for Windows is installed and that it matches the same JDK version as the one installed with Sterling Integrator (wherever it is installed). If Sterling Integrator is installed on the same machine as Gentran:Server for Windows then the correct JDK version will automatically be installed when Sterling Integrator is installed. The system requirements for this installation are Java Runtime Environment/Java Virtual Machine version 1.3.

## Installing the Adapter Client on Gentran:Server for Windows

To install the Gentran:Server Windows adapter client on Gentran:Server for Windows:

1. In the /*<install_dir>*/installed_data/gentrannt/client folder on the computer where Sterling Integrator is installed, locate the GSNTClient.jar file.
2. Move a copy of the GSNTClient.jar file to the *GentranInstallation*/bin folder on the computer where Gentran:Server for Windows is installed.
3. Unpack the GSNTClient.jar file in the *GentranInstallation*/bin folder. GSNTClient.jar contains the following files:
   - GenSrvNtAdapter.jar (Note: do not unpack the GenSrvNtAdapter.jar file!)
   - runremotermi.bat
   - SterlingIntegrator<ver>.dll
   - Manifest.mf
4. Rename the appropriate version of SterlingIntegrator<ver>.dll to SterlingIntegrator.dll. Note that the version of SterlingIntegrator.dll corresponds with the version number of Gentran:Server for Windows.

## Installing Sterling Integrator Gateway in Gentran:Server for Windows

To install a Sterling Integrator gateway in Gentran:Server for Windows:

1. Run **regedit** on the computer where Gentran:Server for Windows is installed and navigate to:
   HKEY_LOCAL_MACHINE\Software\Sterling Commerce\Gentran\Mailbox\ version#\Client\Gateways
2. Right-click the **Client\Gateways** folder and select **New** > **String Value**.
3. In the **String Value Name** box, type STERLINGIntegrator as the name of the gateway.
4. In the **Data** box, type SterlingIntegrator.dll to assign the file to the Sterling Integrator gateway.
   The *xx* in SterlingIntegrator*xx*.dll relates to the version number of Gentran:Server for Windows that you have installed on your computer.
5. Navigate to:
   HKEY_LOCAL_MACHINE\Software\Sterling Commerce\Gentran\Mailbox\ version#\Server\Gateways
6. Right-click the **Server\Gateways** folder and select **New** > **String Value**.
7. In the **String Value Name** box, type STERLINGIntegrator as the name of the gateway.

8. In the **Data** box, type `SterlingIntegrator.dll` to assign the file to the Sterling Integrator gateway.

9. Start the Mailbox service to establish the new gateway settings.

## Creating a Sterling Integrator Mailbox in Gentran:Server for Windows

The SterlingIntegrator.dll file must be placed in the registry before you can create the Sterling Integrator mailbox using the Sterling Integrator gateway.

To create a Sterling Integrator mailbox in Gentran:Server for Windows:

1. In Gentran:Server for Windows, start MailBox Server Manager.

2. Right-click **Mailboxes** and select **Create**.

3. Type a unique name as the name of the mailbox you are creating and click **Next** twice.

4. Click **Yes, use this mailbox as a gateway**, select Sterling Integrator as the type of gateway to use with this mailbox, and then click **Next**.

5. On the summary page, confirm that the information you entered is correct, and then click **Finish**.

   Depending on your system configuration, the mailbox properties window may open.

6. Open the Gentran:Server Mailbox Manager client, right-click **Client\Gateway**, and select **Properties**.

7. Type the content type and subtype of messages and attachments when receiving messages.

8. Click **Edit Recipients** and select the receivers to send the message to.

## Using Remote RMI in Gentran:Server for Windows

To use remote RMI in Gentran:Server for Windows:

1. In the runremotermi.bat file, note the port number at the end of the last line in the file. This is the port number required in the RMI server address during configuration of the Gentran:Server Windows adapter.

   If this port number changes, you must change the configuration of the Gentran:Server for Windows adapter.

2. Run the runremotermi.bat file.

## Configuring the Gentran:Server Windows Adapter

To configure the Gentran:Server Windows adapter, you must specify field settings in Sterling Integrator and in the GPM.

**Sterling Integrator Configuration**

The following table describes the fields used to configure the Gentran:Server Windows 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 | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br>**Note:** See *Using Service Groups*. |
| Host Name (gentranhost) | Name of the computer where Gentran:Server for Windows is installed. |
| Mailbox Name (remoteMailbox) | Name of the mailbox created in Gentran:Server for Windows that this configuration of the adapter will use. **Note:** This mailbox must have Sterling Integrator listed as its gateway. |
| RMI Server Address (remoteAddr) | Address of the RMI server that is installed on the same computer as Gentran:Server for Windows. The format to enter this address is *ipadr:port/rmiServerName*. |
| Start a Business Process When Data is Received? (bootstrap) | Whether the adapter should start a business process. Valid values are Yes and No. Required. |
| Business process (initialWorkFlowId) | Business process to start when the Gentran:Server Windows adapter receives data. Required only if you selected Yes in the Start a Business Process When Data is Received? field. |
| Do not use schedule | If this field is selected, this service does not start a business process and does not run on a schedule. Required for receiving data from Sterling Integrator. |
| Run service based on timer every | Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. |
| Run service daily at | Valid values are the hour and minutes at which to run the service, daily. Indicate whether you want the service to run at startup. |
| Run service weekly on | Valid values are the day of the week, the hour, and the minutes at which to run the service. Indicate whether you want the service to run at startup. |

**GPM Configuration**

The following table describes the fields used to configure the Gentran:Server Windows adapter in the GPM:

| Field | Description |
|---|---|
| Config | Name of the adapter configuration. |

| Field | Description |
|-------|-------------|
| Action | Action that this configuration of the adapter performs. Valid values are Receive from Gentran and Send to Gentran. |

# Chapter 95. GXS ICS FTP Adapter

The following table provides an overview of the GXS (Global Exchange Services) ICS (Interchange Services) FTP adapter:

| | |
|---|---|
| System name | GXS ICS FTP Adapter |
| Graphical Process Modeler (GPM) category | None |
| Description | The GXS ICS FTP Adapter is used to send and collect documents from the GXS Interchange Services (ICS) mailbox via the TCP/IP FTP Gateway. |
| Business usage | This adapter can be used to exchange documents with the Trading Partner that uses the GXS Interchange Services mailbox. |
| Usage example | An internal service invokes a business process that sends a trading document to the trading partner mailbox in the GXS VAN. The BPML can be invoked manually or by an internal service. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | FTP Client Adapter |
| Application requirements | None |
| Initiates business processes? | Yes, this adapter initiates business processes when it is in the Collect mode. |
| Invocation | This adapter is invoked by an internal service. |
| Business process context considerations | The configuration parameters and the outgoing document are picked up by the GXS ICS FTP adapter in the workflow context. |
| Workflow Context Considerations | You must be familiar with the internal service (workflow context/business process parameters) that invoked this adapter. Workflow parameters are passed into the internal service and business process parameters are the values specified in the business process code. |
| Returned status values | Returned status values:<br>• 0 – Success<br>• 1 – Error |
| Restrictions | None |
| Persistence level | System default |
| Testing considerations | To test an instance of the GXS ICS FTP adapter, create a GXS_ICS_Adapter by clicking **Deployment**->**Services**->**Configuration** and choosing **GXS ICS FTP Adapter**. The most frequent problems encountered are:<br>• Incorrect parameters are set up.<br>• The adapter is not active.<br><br>If you get an error with any of these conditions, check with your System Administrator. |

## Implementing the GXS ICS FTP Adapter

To implement the GXS ICS FTP adapter, complete the following tasks:

1. Activate your license for the GXS ICS FTP adapter. See *An Overview of Implementing Services*.
2. Create an GXS ICS FTP adapter configuration. See *Creating a Service Configuration*.
3. Configure the adapter. See *Configuring the GXS ICS FTP Adapter*.
4. Create and enable a business process that includes the GXS ICS FTP adapter.
5. Test the business process and the adapter.
6. Run the business process.

## Configuring the GXS ICS FTP Adapter

To configure the GXS ICS FTP adapter, you must specify field settings in Sterling Integrator. See *Creating a Service Configuration*.

The following table describes the fields used to configure the GXS ICS FTP 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 | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** See *Using Service Groups*. |
| Default GXS ICS FTP Server | DNS name or IP address of the GXS ICS FTP server that Sterling Integrator is to connect to. This is any valid DNS name or a dotted decimal address. The default value (myhost) is shown as an example. Required |
| Default GXS ICS FTP Server Port | Port number of the GXS ICS FTP server. This is typically port number 21, but it can be another number depending on the FTP server settings. Default is 21. Required. |
| User Name | User login name to connect to the GXS ICS FTP server. Alphanumeric and case-sensitive. Required. |
| Password | Password used for authentication when connecting to the GXS ICS FTP server. Case-sensitive. Required. The password will be obscured (encrypted) and will appear as all asterisks. |
| SMIME Encryption User Certificate | Not used |
| SMIME Decryption Certificate | Not used |

| Field | Description |
|---|---|
| SSL | Flag that determines Secure Socket Layering (SSL) socket negotiation. Required. Valid values: <br> • SSL_IMPLICIT – Indicates that the GXS ICS FTP server expects and requires SSL to happen automatically at the time of connection. <br> • SSL_EXPLICIT – Indicates that the GXS ICS FTP client requests SSL and a secure connection is negotiated. <br> • None – Indicates that the connection will not use SSL. |
| Clear Control Channel | Indicates if information that travels across the control channel should be clear. Required. Valid values: <br> • Yes – Information will be clear <br> • No – Information does not need to be clear |
| Key Certificate Passphrase | Not used |
| Cipher Strength | Level of encryption to be applied to the data that flows through the socket connection. Optional. Valid values: <br> • ALL – All cipher strengths are supported. <br> • WEAK – Often required for international trade, because government regulations prohibit STRONG encryption from being exported. <br> • STRONG – Highest level of encryption. <br> Default is STRONG. |
| Key Certificate (System Store) | PrivateKeys / Public Certificates that are signed by the Trading Partner Trusted Certificate Authority. In process data, this parameter is displayed as an object ID. Select a valid key certificate from the list of those already stored inside Sterling Integrator. Required if the GXS ICS FTP server requires client authentication using SSL. |
| CA Certificates | Trusted Certificate Authority public certificate. In process data, this parameter is displayed as an object ID. Select a valid CA certificate from the list of those already stored inside the application. Required for server authentication using SSL. |
| Type of operation | Type of operation to performValid values: <br> • Collect – Collect messages from the GXS ICS mailbox <br> • Submit – Submit mailbox messages to the GXS ICS mailbox |
| Select the type of message to collect | Type of message to collect from the GXS ICS mailbox. Required if **Type of operation** is set to Collect. Valid values: <br> • Mailbox messages – Collect mailbox messages. <br> • Report – Retrieve a report. <br> Default is Mailbox messages. |

| Field | Description |
|---|---|
| Select the report you want to retrieve | Type of report to retrieve. Required if **Type of operation** is set to Collect and **Select the type of message to collect** is set to Report. Valid values:<br>• Sender Status Report<br>• Sender Summary<br>• Sender Unretrieved<br>• Sender Error<br>• Receiver Status<br>• Receiver Summary<br>• Third Party Reject<br>• Trading Pair Trxn Detail |
| Specify the message retrieval criteria | Whether to collect all messages or only those meeting the criteria specified in **Sender or Receiver address** or in **Match this ILog number**. Required if **Type of operation** is set to Collect and **Select the type of message to collect** is set to Mailbox messages. Valid values:<br>• Download all interchanges<br>• Download interchanges that match the criteria specify below<br>Default is Download all interchanges. |
| Sender or Receiver address | Address on the GXS ICS FTP server to download messages from. Optional. |
| Match this ILog number | ILog number to download messages from the GXS ICS FTP server under. Optional. |
| Specify the message download option | How to save the downloaded documents. Required if **Type of operation** is set to Collect and **Select the type of message to collect** is set to Mailbox messages. Valid values:<br>• Download all interchanges into a single document<br>• Download each interchanges as an individual document<br>Default value is Download all interchanges into a single document |
| Business Process | Name of the business process you want this adapter to start. Required if **Type of operation** is set to Collect. |
| Run As User | Applies to the scheduling of the business process. The Run As User field only displays as an option if **Type of operation** is set to Collect. Type the user ID to associate with the schedule, or click the<br><br>icon and select a user ID from the list. Valid values:<br><br>Any valid Sterling Integrator user ID<br>**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. |
| 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 | Information about scheduling the collection action. Required if **Type of operation** is set to Collect. Valid values: |
| | • Do not use schedule If this field is selected, the adapter does not start a business process and does not run on a schedule. |
| | • Run based on timer Valid values are the hour and minutes at which to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any schedule exclusions or date exclusions. Indicate whether you want the adapter to run at startup. |
| | • Run daily Valid values are the hour and minutes at which to run the adapter, daily. If you choose to select a time interval, the valid values are the hour and minute for the interval. Add or delete selections as necessary. Specify any date exclusions. Indicate whether you want the adapter to run at startup. |
| | • Run based on days of the week Valid values are the day of the week, the hour, and the minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |
| | • Run based on days of the month Valid values are the day of the month, hour, and minute that specify when to run the adapter. If you choose to select a time interval, the valid values are the hours and minutes for the intervals. Add or delete selections as necessary. Specify any date exclusions. |

## Before Using the GXS ICS FTP Adapter in a Business Process

If the GXS ICS FTP server that is communicating with Sterling Integrator uses SSL communication, the CA certificate object ID or system key certificate object ID (CACert or Keycert) files must be saved before the GXS ICS FTP adapter can be used in a business process. The Sterling Integrator system administrator must save these certificates and private key in the Sterling Integrator database.

## Business Process Configuration

No GXS ICS FTP adapter parameters are configured with the Graphical Process Modeler (GPM). All parameters must be configured in Sterling Integrator.

## ILOG Number

A unique ILOG number is generated per file upload to the GXS ICS FTP server. The adapter captures the ILOG number and stores it in the process data. This enables the downstream workflow to act on to this information. The ILOG number is stored in the following format:

```
<ProcessData>
  <GXSResponses>
     <IlogNumber>7311258167</IlogNumber>
  </GXSResponses>
</ProcessData>
```

## Certificate Authority (CA) Certificate

Currently, a GXS Server accepts any valid Verisign class 3 CA certificate. You are required to check these certificates in to the application (using Trading Partners > Digital Certificates > CA). One of these certificates must be selected when you configure the GXS ICS FTP adapter instance. Customers are encouraged to check with GXS to ensure that the GXS Server is still using Verisign as its CA prior to using the GXS ICS FTP adapter.

## Example Business Process

The following sample BPML invokes an instance of the GXS ICS FTP adapter.

```
<process name="Test_bpml">
  <sequence name="optional">
    <operation name="Invoke an instance of GXS ICS FTP adapter ">
    <participant name="GXS_ICS_FTP_adapter"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
         <assign to="." from="*"></assign>
      </input>
     </operation>
 </sequence>
</process>
```

# Chapter 96. Human Interaction Document Loader Service

The Human Interaction Document Loader service retrieves a document previously saved in the Sterling Integrator database by the Human Interaction Event service. After the document is retrieved, the Human Interaction Document Loader service loads the document into the business process as the primary document. The following table provides an overview of the Human Interaction Document Loader service:

| System name | HumanInteractionDocumentLoader |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Web Extensions |
| Description | Uses a valid MinedDataId of a previously saved document to retrieve the document from the Sterling Integrator database.<br>**Note:** The MinedDataId is created when a document is added to the database using the Human Interaction Event service, and corresponds to the value in the MINED_DATA_ID column of the WEBX_MINED_DATA table. |
| Business usage | A business process uses the Human Interaction Event service to save a document requiring approval in the database and then sends out an e-mail notification to the approver. The e-mail notification includes the URL with the Web template, along with the MinedDataId, for retrieving and displaying the document. The Web template includes a call to a business process that uses the Human Interaction Document Loader service to retrieve the document and the B2B HTTP Server adapter for displaying the document in the Web browser for approval or rejection.<br>**Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*. |
| Usage example | The individual responsible for approving a purchase order does the following: Receives an e-mail notification with a URL link for displaying the purchase order. Launches a Web browser and specifies the URL link. Approves or rejects the purchase order displayed in the Web browser. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |

| Related services | Designed to work with: |
| --- | --- |
| | • Human Interaction Query service |
| | • B2B Lookup System service |
| | • B2B HTTP Server adapter |
| | • HTTP Server adapter |
| | • Human Interaction XForms service |
| | **Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*. |
| 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: |
| | • WorkFlowContext.SUCCESS: Success, with no errors. |
| | • WorkFlowContext.ERROR NoMinedDataIDException: Caught NoMinedDataIDException in DocumentLoaderService.processData() |
| Restrictions | None |
| Persistence level | None |
| Testing considerations | None |

### How the Human Interaction Document Loader Service Works

Use the Human Interaction Document Loader service to retrieve documents from the Sterling Integrator database. The following sections describe a business scenario and a sample solution using the Human Interaction Document Loader service.

### Business Scenario

Your company receives purchase orders in XML format from a trading partner. All purchase orders exceeding $1,000.00 need to be approved. You want to view a purchase order requiring approval in the Web browser.

### Business Solution Example

The approach used to solve this business scenario includes creating a business process that includes:

• A Human Interaction Document Loader service to retrieve the purchase order requiring approval
• A B2B HTTP Server adapter or HTTP Server adapter to display the purchase order in the Web browser.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

## Business Solution Example Dependencies

The following dependencies must be met to run the example in this section and display the purchase order in a Web browser:

- Run a business process that accepts a purchase order as input and uses the Human Interaction Event service to store the purchase order in the Sterling Integrator database and mark it as requiring approval. The Human Interaction Event service must also be configured to invoke a business process that constructs and sends an e-mail notification containing the URL for displaying the purchase order to the approver. The URL must contain the name of the Web template, along with the document ID for the purchase order requiring approval. See the business scenario and business solution example for the B2B Lookup System service for an example business process.
- Deploy a Web template in Sterling Integrator that invokes the business process in this example and displays the purchase order in the Web browser.
- Create a configuration of the HTTP Respond service that specifies the URI used to invoke the business process created in this example from the Web browser.

This business solution example focuses only on the Human Interaction Document Loader service functionality.

## GPM Example

The following example shows a solution to the business scenario using the GPM.

**Note:** This example assumes that the Web template used to display the purchase order is deployed in Sterling Integrator.
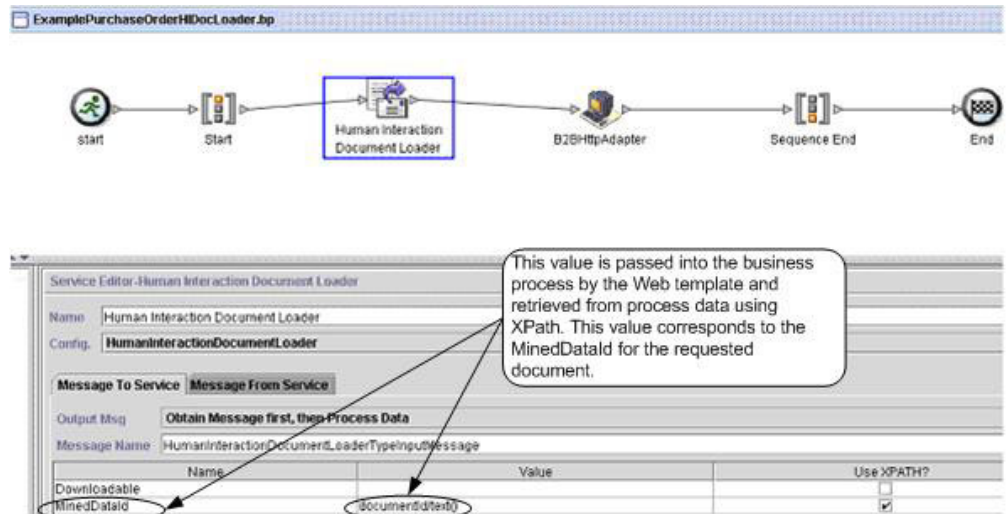
The Human Interaction Event service (not shown) generated an e-mail notification to the approver after the purchase order was added to the database. The approver uses this URL to display the purchase order requiring approval in the Web browser. For example:

```
http://siHostName:siport/websuite/xfm/
ExamplePurchaseOrderFromHTTPRequest.xfm?documentId=7736bd:fb78c70638:-77df
```

where:
- ExamplePurchaseOrderFromHTTPRequest.xfm is the name of the Web template used to display the purchase order. This Web template is designed to invoke the business process that retrieves the purchase order from the Sterling Integrator database.
- documentId is set to the MinedDataId (7736bd:fb78c70638:-77df) of the requested document in the Sterling Integrator database.

The business process run by the Web template uses the Human Interaction Document Loader service to retrieve from the database the purchase order that corresponds to the documentId, and uses the B2B HTTP Server adapter to display the purchase order in the Web browser.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters*.

After the Human Interaction Document Loader service completes, the purchase order is returned to the business process as the primary document, along with a document named HumanInteractionEvents, which contains the reference data from the Human Interaction Event service for the purchase order. The returned Human Interaction Events document for this example looks like the following:



The returned primary document (purchase order) for this example looks like the following:

```
<Order Id="200">
 <Order_Item Id="400">
  <Order_Id>200</Order_Id>
  <Name>DSL</Name>
  <Quantity>1</Quantity>
  <Price> $1050.00</Price>
 </Order_Item>
 <Order_Item Id="401">
  <Order_Id>200</Order_Id>
  <Name>Cable</Name>
  <Quantity>1</Quantity>
  <Price> $1500.00</Price>
 </Order_Item>
</Order>
```

The primary document is then sent to the B2B HTTP Server adapter for display in the Web browser.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters*. **For example, you might have a Web template that displays the purchase order in the following HTML format:**



The approver can choose to approve or reject the purchase order.

## Business Process Modeling Language (BPML) Example

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

```
<process name="ExamplePurchaseOrderHIDocLoader">
<sequence name="Start">
<operation name="Human Interaction Document Loader">
<participant name="HumanInteractionDocumentLoader"/>
<output message="HumanInteractionDocumentLoaderTypeInputMessage">
 <assign to="MinedDataId" from="documentId/text()"/>
 <assign to="." from="*"/>
 </output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
<operation name="HTTP Respond Service">
      <participant name="HttpRespond"/>
      <output message="HttpRespondServiceInputMessage">
        <assign to="." from="*"></assign>
```

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

## Implementing the Human Interaction Document Loader Service

To implement the Human Interaction Document Loader service, complete the following tasks:

1. Activate your license for the Human Interaction Document Loader service. See *Installing Sterling Integrator*.

2. Create a Human Interaction Document Loader service configuration. See *Managing Services and Adapters*.

3. Use the Human Interaction Document Loader service in a business process. For information, see *Human Interaction Document Loader Service Business Process Usage*.

## Human Interaction Document Loader Service Business Process Usage

The following screen shows a graphical view of the GPM parameters for the Human Interaction Document Loader service. There are no fields to be configured on the Message From Service tab.



The following example shows the corresponding BPML parameters for the Human Interaction Document Loader service GPM parameters.

```
<process name="ExamplePurchaseOrderHIDocLoader">
<sequence name="Start">
<operation name="Human Interaction Document Loader">
<participant name="HumanInteractionDocumentLoader"/>
<output message="HumanInteractionDocumentLoaderTypeInputMessage">
 <assign to="Downloadable">YES</assign>
 <assign to="MinedDataId" from="documentId/text()"/>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

The following table describes the fields used to configure the Human Interaction Document Loader service in the GPM. The values specified for these parameters are the input to the Human Interaction Document Loader service from the business process.

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| Downloadable | Whether a file download box displays when trying to download files. This parameter applies to Web Extension applications that provide a means to download, save, or display the content of a file saved in a workflow. Valid values: |
| | • Yes – Always displays the file download box each time you download a file to your client computer, regardless of file type. This gives you the option of saving the file to disk or displaying the file contents in the browser regardless of the file type. |
| | • No – Determines whether or not to display a file download box based on the file content. For example, if the downloaded file is in XML format, the browser displays the raw XML in the browser rather then prompting you to save the file. If the downloaded file is text format, the browser prompts you to either display or save the file to disk. |
| MinedDataId | Unique identifier that is associated with a document within a business process and corresponds to the value in the MINED_DATA_ID column of the WEBX_MINED_DATA table in the Sterling Integrator database. This value is populated when a document is added to the database using the Human Interaction Event service. Valid value is a MinedDataId that exists in the WEBX_MINED_DATA table. <br> **Note:** If you want to reuse the business process and service, you should not hardcode this value. Instead, use an XPath expression to set this parameter. |

## Output from Service to Business Process

The following table describes the output from the Human Interaction Document Loader service to the business process:

| Returned Output | Description |
|---|---|
| HumanInteractionEv ents/@SCIObjectID document | The name of the document, along with the document SCI Object ID, returned by the Human Interaction Document Loader service. This document contains the reference data from the Human Interaction Event service for the specific document retrieved from the Sterling Integrator database. |

# Chapter 97. Human Interaction Event Service

The Human Interaction Event service flags business processes that require human action and creates references to the business process data to await human action.

The Human Interaction Event service:

- Is the service that starts human interaction at one or more steps in the business process.
- Handles the updating and deletion of mined data, which is data pulled from the business process and written to the Sterling Integrator internal database for future use through demining. Demining is the action of retrieving data from the internal database for display in a Web browser.
- Is used at any point in a business process where human interaction is needed.

The following table provides an overview of the Human Interaction Event service:

| System name | HumanInteractionEvent |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Web Extensions |
| Description | The Human Interaction Event service:<br><br>• Adds documents requiring Human Interaction to the WEBX_MINED_DATA table in the Sterling Integrator database. In addition to adding the document, reference data about the document and the current state of the business process used to invoke the Human Interaction Event service is added to the database.<br>• Updates documents, along with the document reference data in the database.<br>• Deletes documents from the database, along with the document reference data.<br><br>This service can also suspend a business process until certain actions take place (such as approval of a document) and then resume processing. |

| | |
|---|---|
| Business usage | A business process uses the Human Interaction Event service to save documents requiring approval in the database and then send out an e-mail notification to the approver. Another business process might include the following services and adapters to work with the documents saved in the database:<br><br>• Human Interaction Query service to retrieve all documents requiring approval from the database<br><br>• Human Interaction XForms service and B2B HTTP Server adapter to display the results from the Human Interaction Query service to the Web browser for user selection<br><br>**Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.<br><br>• Human Interaction Document Loader service to retrieve the selected document from the database and pass the document to the Human Interaction XForms service for displaying in the Web browser for approval |
| Usage example | An organization may want to monitor all purchase orders, requiring approval for any PO request that exceeds $1000.00. The business process associated with the submission of a PO could be configured to send the PO request to the employee's manager, suspending the business process until the approval is complete. After the PO is approved or rejected, the business process completes the steps to either submit the PO or send a rejection back to the employee. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Designed to work with:<br><br>• Human Interaction Query service<br><br>• Human Interaction Document Loader service<br><br>• Human Interaction XForms service |
| Application requirements | None |
| Initiates business processes? | The user can specify the WebSuiteEmailNotif business process on the Human Interaction Event service NotificationBPName parameter to invoke this business process for generating an e-mail notification. The WebSuiteEmailNotif business process comes predefined with Sterling Integrator. The user can also create a customized business process that sends e-mail notifications in a specific format and invokes the business process using the NotificationBPName parameter. |
| Invocation | Runs as part of a business process. |
| Business process context considerations | None |

| Returned status values | Returned status values: |
|---|---|
| | • WorkFlowContext.SUCCESS: Success, with no errors. |
| | • WorkFlowContext.ERROR IllegalArgumentException: Caught IllegalArgumentException in EventService.processData() |
| | • WorkFlowContext.ERROR IllegalStateException: Caught IllegalStateException in EventService.processData() |
| | • WorkFlowContext.ERROR SQLException: Caught SQLException in EventService.processData() |
| Restrictions | None |
| Testing considerations | None |

## How the Human Interaction Event Service Works

Use the Human Interaction Event service to save business process data requiring human interaction in the Sterling Integrator database. The following sections describe a business scenario and a sample solution for using the Human Interaction Event service.

## Business Scenario

Your company receives a purchase order in XML format from a trading partner. The purchase order needs to be approved prior to further processing.

The approach used to solve this business scenario includes the creation of a business process that does the following:

• Accepts the purchase order as input.
• Includes a Human Interaction Event service that saves a reference to the purchase order, the reference data about the purchase order, and the current state of the business process data in the Sterling Integrator database and marks the business process as requiring human interaction. The reference data for this example includes the following information:
  – The document type (purchase order)
  – The Sterling Integrator user account of the person sending the purchase order
  – The Sterling Integrator user account of the person who is to receive the purchase order (approver)
  – The state of the purchase order (Awaiting Approval)
  – The status of the purchase order (Unread)
  – The Web template used to display the purchase order in the Web browser

The Human Interaction Event service is also configured to invoke the WebSuiteEmailNotif business process for sending an e-mail notification to the approver. The e-mail notification includes the URL used to display the purchase order requiring approval in the Web browser.

**Note:** You must have a valid, checked-in Web template to display an item requiring human interaction in the Web browser.

This business solution example focuses only on the Human Interaction Event service functionality.

## GPM Example

The following example shows a solution to the business scenario using the GPM. The purchase order is input to the business process and becomes the primary document.

(Screen 1 of 2)



(Screen 2 of 2)



The Assign statement is used to assign the purchase order number from the primary document to process data. The Human Interaction Event service uses the purchase order number in process data to set its ReferenceId parameter.

After the Human Interaction Event service completes, a row is added to the Sterling Integrator WEBX_MINED_DATA database table containing the reference data about the purchase order and a reference to the actual purchase order document. In addition, an e-mail notification is sent to the approver that includes the URL for displaying the actual purchase order. For example:

```
STERLING Integrator - Web Suite Email Notification

You have received a new Purchase Order in your Web Suite Inbox.

To:               ReceiverUserAccount
From:             SenderUserAccount
RE:               Purchase Order - 200
Location:         Inbox
Timestamp:        2004-12-17T18:49:55.000Z
Expires (minutes):

http://00.000.0.000:00000/websuite/xfm/
ExamplePurchaseOrderFromHTTPRequest.xfm?documentId=1446e4:fb582af649:-5dcc
```

The WebSuiteEmailNotif business process, which was set with the Notification BPName parameter, is used to construct this URL.

**Note:** The Web template specified in the URL should be designed to invoke a business process that uses the Human Interaction Document Loader service to load the purchase order (identified by its documentId in the URL) from the Sterling Integrator database prior to displaying the Web page.

The approver specifies the URL in the Web browser to display the purchase order requiring approval. The approver could alternatively log into the Sterling Integrator Web Suite application to view (in their inbox) a list of documents requiring approval.

## Business Process Modeling Language (BPML) Example

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

```
<process name="ExamplePurchaseOrderHIEventWithMail">
<sequence name="Start HI Event Service">
<assign name="Assign" to="poNumber" from="DocToDOM(PrimaryDocument)/
     @Id" append="true"/>
<operation name="Human Interaction Event">
<participant name="HumanInteractionEvent"/>
<output message="HumanInteractionEventTypeInputMessage">
 <assign to="DocumentType">Purchase Order</assign>
 <assign to="FromAccount">SenderSterlingIntegratorUserAccount</assign>
 <assign to="NotificationBPName">WebSuiteEmailNotif</assign>
 <assign to="Operation">ADD</assign>
 <assign  to="ReferenceId" from="number(/ProcessData/poNumber/@Id)"/>
 <assign to="Status">Awaiting Approval</assign>
 <assign to="StorageArea">Inbox</assign>
 <assign to="SuspendAsWaiting">NO</assign>
 <assign to="SystemAccount">ReceiverSterlingIntegratorUserAccount</assign>
 <assign to="TemplateName">ExamplePurchaseOrderFromHTTPRequest</assign>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

See *Usage Examples* for additional examples using the Human Interaction Event service.

## Implementing the Human Interaction Event Service

To implement the Human Interaction Event service, complete the following tasks:

1. Activate your license for the Human Interaction Event service. See *Installing Sterling Integrator*.
2. Create a Human Interaction Event service configuration. See *Managing Services and Adapters*.
3. Use the Human Interaction Event service in a business process.

## Human Interaction Event Service Business Process Usage

The following screen shows the graphical view of the GPM parameters for the Human Interaction Event service. There are no fields to be configured on the Message From Service tab.



The following example shows the corresponding BPML parameters for the Human Interaction Event service GPM parameters.

```
<process name="ExampleHIEventGPMParams">
<sequence name="Start HI Event Service">
<operation name="Human Interaction Event">
<participant name="HumanInteractionEvent"/>
<output message="HumanInteractionEventTypeInputMessage">
 <assign to="ArchiveFlag">1</assign>
 <assign to="ArchiveHours">24</assign>
 <assign to="DocumentType">Purchase Order</assign>
 <assign to="FromAccount">SendingTradingPartnerSystemAccountID</assign>
 <assign to="GroupName">SendingTradingPartnerSecurityPermissionGroup</assign>
 <assign to="IdentityName">SendingTradingPartnerName</assign>
 <assign to="MergeDataOnResume">NO</assign>
 <assign to="MergeDataOnUpdate">NO</assign>
 <assign to="NotificationBPName">UserDefinedBPForSendingEmailNotifications</assign>
 <assign to="Operation">ADD</assign>
 <assign to="ParentAccount">ManagerOfTheDocumentSender</assign>
 <assign to="ReferenceId" from="number(/ProcessData/poNumber/@Id)"/>
 <assign to="State">PendingApproval</assign>
 <assign to="Status">Unread</assign>
 <assign to="StorageArea">Inbox</assign>
 <assign to="SuspendAsWaiting">NO</assign>
 <assign to="SystemAccount">ReceiverCompanyUserAccount</assign>
 <assign to="TemplateName">ExamplePurchaseOrders</assign>
 <assign to="Timeout"/>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
```

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

The following table describes the fields used to configure the Human Interaction Event service in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| ArchiveHours | The number of hours the mined data record should remain in the Web Extension table before it is archived or purged. Valid value is a positive integer. Optional. If blank, the time span from the business process is used, if it has been specified. If the time span from the business process has not been specified, the system default is used. |
| ArchiveFlag | The archive method that should be used. Optional. Valid values:<br>• Archive – Archives the data in the Web Extension table.<br>• Purge – Deletes the data from the Web Extension table.<br>If left blank, the archive method from the business process is used. |
| DocumentType | Type of data that requires human interaction. Valid value is any alphanumeric string. Examples include Purchase Order, Invoice, and Remittance Advice.<br>**Note:** The value specified for this parameter is the name of the document that appears under Document Type in the user's inbox when using the Sterling Integrator Web Suite to display documents. |
| FromAccount | Trading partner system user account ID (as defined in Sterling Integrator) of the trading partner who sent the document. This account is also associated with the trading partner's trading profile. |
| GroupName | Name of the security permission group to which the trading partner who sent the document belongs. |
| IdentityName | Name of the trading partner who sent the document. This name is associated with the FromAccount information listed in this service. This parameter allows for searching records pertaining to a specific company. |
| MergeDataOnResume | Merges any changes made to the Web template with the existing instance data when the business process is taken out of the waiting state. Valid values:<br>• YES – Causes the documents and instance data to be merged when the business process is resumed.<br>• NO – Overlays the business process being resumed (taken out of a waiting state) with the changed documents and instance data. Default. |

| Field | Description |
|-------|-------------|
| MergeDataOnUpdate | Merges any changes made to the Web template with the existing instance data. Valid values :<br>• YES – Default.<br>• NO – Update only the status or state of an item and not the document contents.<br><br>The parameter affects processing only when the Operation parameter is set to UPDATE. |
| NotificationBPName | Business process (for example, the WebSuiteEmailNotif business process) runs by the Human Interaction Event service with the purpose of generating an e-mail notification to the individual who receives the document. The WebSuiteEmailNotif generates an e-mail notification based on the e-mail address associated with the Sterling Integrator user account (SystemAccount) set up for the trading partner receiving the document and the user account (FromAccount) set up for the trading partner who sent the document. The WebSuiteEmailNotif business process comes predefined with Sterling Integrator. Alternatively, you can create and specify a customized business process that sends e-mail notifications in a specific format. If you choose this method, you should model the WebSuiteEmailNotif business process. Valid values are any checked-in business process from the list that is designed to generate e-mail notifications. The default value is Not Applicable.<br>**Note:** If you are using a business process that looks up the e-mail addresses from the user accounts specified with the SystemAccount and FromAccount parameters and if the Sterling Integrator user account does not contain a valid e-mail address, the associated SMTP Send adapter fails and the e-mail notification is not sent. |
| Operation | The requested action for the document. Required. Valid values:<br>• ADD – Add a new document along with its reference data.<br>• DELETE – Delete an existing document along with its reference data.<br>• UPDATE – Update an existing document along with its reference data.<br><br>**Note:** You must specify the MinedDataId parameter as an input parameter to the Human Interaction Event service for the DELETE and UPDATE operations. Use the Advanced Editor or BPML to specify the MinedDataId parameter. See *Updating or Deleting a Document in the Application Database* for more information about the MinedDataId and examples for specifying the parameter. |
| ParentAccount | Account name for the manager of the From Account—the document sender. This user account name is associated with the FromAccount information in this service. |

| Field | Description |
|---|---|
| ReferenceID | Document identifier. Valid value is any alphanumeric string, such as a purchase order number or an invoice number.<br>**Note:** The value specified for this parameter is displayed as the Identifier in the user's inbox when using the Sterling Integrator Web Suite to display documents. |
| State | Document process stage. Valid value is any alphanumeric string, such as Approved, Pending, or Rejected. If no state is entered, the default is Active. |
| Status | Status of the document. Valid value is any alphanumeric string, such as Read or Unread.<br>**Note:** The value specified for this parameter is displayed as the Status in the user's inbox when using the Sterling Integrator Web Suite to display documents. |
| StorageArea | Virtual location to store the data waiting for the person to take action on it. Valid value is any alphanumeric string, such as Inbox, Outbox, and Drafts.<br>**Note:** The value specified for this parameter corresponds to a folder name used to display documents from this storage area in the Sterling Integrator Web Suite. |
| SuspendAsWaiting | Setting for specifying whether to place the business process into a waiting status until human interaction is complete. Required if Operation field is set to ADD or UPDATE. Valid values:<br><br>• Yes – Business process is placed in a wait state. The business process continues waiting until an action is taken on the document (such as updating by another instance of the Human Interaction Event service) or until the value specified with the Timeout parameter expires. See *Placing a Business Process in a Waiting State for Human Interaction* for an example.<br><br>• No – Business process continues processing and the document becomes available for viewing. |
| SystemAccount | Trading partner user account ID (as defined in Sterling Integrator) of the person who must interact with the business process.<br>**Note:** The document is stored in this user's storage area (such as the inbox) in the Sterling Integrator Web Suite. |
| TemplateName | Name of the Web template to use for viewing the data. Select the Web template name from the list of deployed Web templates. Required if you are using the Sterling Integrator Web Suites to view documents.<br>**Note:** If you are using a custom Web application for viewing data in the Sterling Integrator database, you can either specify the Web template with the Human Interaction Event service or with the Human Interaction XForms service. |

| Field | Description |
|---|---|
| Timeout | Amount of time, in minutes, before the waiting status expires, causing the business process to fail with an advanced status of Human Interaction Timeout Event. Timeout is recognized only if the SuspendAsWaiting field is set to YES.<br>**Note:** The Timeout parameter also is dependent on the BPExpirator. The BPExpirator schedule looks for expired business processes based on the time set in the scheduler and resumes the business process with an advanced status. The default schedule time is 15 minutes. |

## Usage Examples

The following sections contain additional examples for using the Human Interaction Event service.

### Placing a Business Process in a Waiting State for Human Interaction

The example in this section describes a scenario that uses the Human Interaction Event service to save a reference to the business process data in the database and place the business process in a wait state until the purchase order is approved. The Human Interaction Event service is also configured to invoke the WebSuiteEmailNotif to send an e-mail notification to the approver. After the purchase order is approved, the business process resumes and continues with the next step. In this scenario, the next step is to invoke a subprocess for further processing of the purchase order.

The following example illustrates this scenario using the GPM.
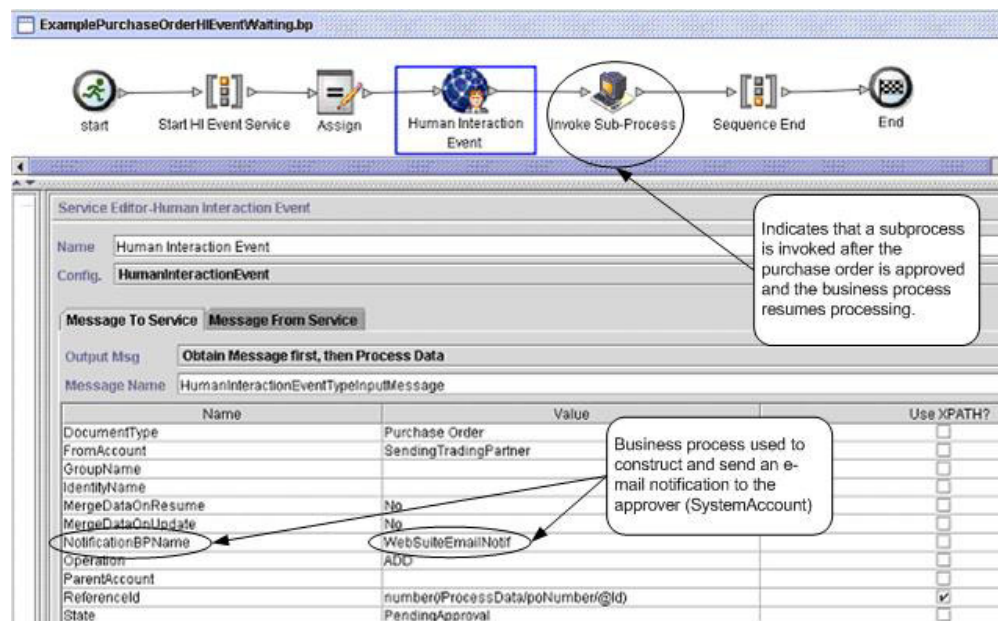


Screen 1 of 2
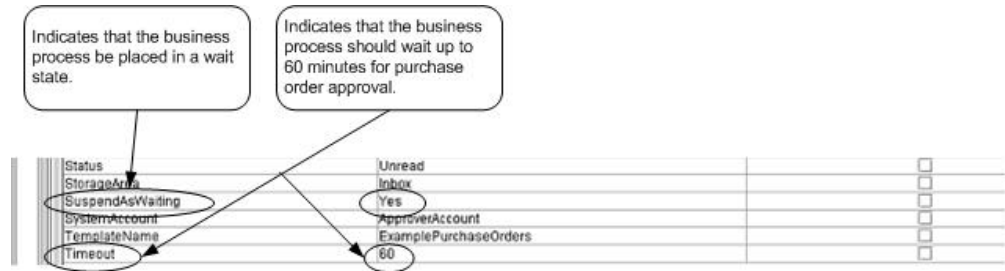
Indicates that the business process be placed in a wait state.

Indicates that the business process should wait up to 60 minutes for purchase order approval.

| Status | Unread | |
| StorageArea | Inbox | |
| SuspendAsWaiting | Yes | |
| SystemAccount | ApproverAccount | |
| TemplateName | ExamplePurchaseOrders | |
| Timeout | 60 | |

(Screen 2 of 2)

The Assign statement is used to assign the purchase order number from the primary document to process data. The Human Interaction Event service uses the purchase order number in process data to set its ReferenceId parameter. After the document is added to the database, you can use the following services to complete the approval process:

- Human Interaction Query service to retrieve a list of purchase order references awaiting approval
- Human Interaction Document Loader service to retrieve the actual purchase order associated with the reference returned from the Human Interaction Query service
- Human Interaction XForms service to display the purchase order in the browser for necessary updates and approval
- Human Interaction Event service to update the database with the changes made to the purchase order and to change the status of the purchase order to Approved. After updating the purchase order, the originating business process resumes processing and starts the subprocess. See *Updating or Deleting a Document in the Application Database* for more information.

The following example illustrates the business process using BPML:

```
<process name="ExamplePurchaseOrderHIEventWaiting">
<sequence name="Start HI Event Service">
<assign to="poNumber" from="DocToDOM(PrimaryDocument)/@Id" append="true"/>
<operation name="Human Interaction Event">
<participant name="HumanInteractionEvent"/>
<output message="HumanInteractionEventTypeInputMessage">
 <assign to="DocumentType">Purchase Order</assign>
 <assign to="FromAccount">SendingTradingPartner</assign>
 <assign to="MergeDataOnResume">NO</assign>
 <assign to="MergeDataOnUpdate">NO</assign>
 <assign to="NotificationBPName">WebSuiteEmailNotif</assign>
 <assign to="Operation">ADD</assign>
 <assign to="ReferenceId" from="number(/ProcessData/poNumber/@Id)"/>
 <assign to="State">PendingApproval</assign>
 <assign to="Status">Unread</assign>
 <assign to="StorageArea">Inbox</assign>
 <assign to="SuspendAsWaiting">YES</assign>
 <assign to="SystemAccount">ApproverAccount</assign>
 <assign to="TemplateName">ExamplePurchaseOrders</assign>
 <assign to="Timeout">60</assign>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
<operation name="Invoke Sub-Process">
<participant name="InvokeSubProcessService"/>
```

```
<output message="InvokeSubProcessServiceTypeInputMessage">
 <assign to="INVOKE_MODE">ASYNC</assign>
 <assign to="WFD_NAME">ExampleHIEventSubProcess</assign>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

For more information, see:

- *Human Interaction Query Service* for a business scenario describing how to use this service to retrieve the purchase order reference data from the Sterling Integrator database.
- *Human Interaction Document Loader Service* for a business scenario describing how to use this service to retrieve the actual purchase order from the Sterling Integrator database.
- *Human Interaction XForms Service* for a business scenario describing how to use this service to display a list of documents.

## Updating or Deleting a Document in the Sterling Integrator Database

You must specify the MinedDataId parameter as an input parameter to the Human Interaction Event service when updating or deleting a document in the database. The MinedDataId parameter corresponds to the value in the MINED_DATA_ID column of the WEBX_MINED_DATA table. This value is populated when a document is added to the database using the Human Interaction Event service.

Use the Human Interaction Query service to retrieve the reference data for documents. The reference data includes the MinedDataId for each document. The MinedDataId corresponds to the listItem objectId attribute returned by the Human Interaction Query service.

You can specify these fields using the Advanced Editor option in the GPM Service Editor or by using BPML.

The following example illustrates specifying the MinedDataId parameter in the GPM. This example assumes that the MinedDataId value was loaded into process data (not shown).

ExamplePurchaseOrderHIQueryAndHIXformSTUB.bp

```
start    Start1    Human Interaction    Sequence End    End
                        Event
```

Service Editor -Human Interaction Event

Name: Human Interaction Event

Config. HumanInteractionEvent

Message To Service | Message From Service

Output Msg    Obtain Message first, then Process Dat

Message Name  HumaninteractionEventTypeInputMessag

| Name | |
|------|--|
| ArchiveHours | |
| DocumentType | Purchase |
| FromAccount | Joe Suppli |
| GroupName | |
| IdentityName | |
| MergeDataOnResume | |
| MergeDataOnUpdate | |
| NotificationBPName | [Not Applicable] |
| Operation | UPDATE |
| ParentAccount | |

Advanced Editor : Human Interaction Event

| Name | Value | Use XPATH? |
|------|-------|------------|
| MinedDataId | string(/ProcessData/list/listItem/@objectId) | ✓ |

Add
Delete

In the Advanced Editor, specify the MinedDataId parameter and its value.

OK    Cancel

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

Advanced

The following example illustrates the same business process using BPML. This example assumes that the MinedDataId value was loaded into process data (not shown).

```
<process name="ExamplePurchaseOrderHIQueryAndHIXformSTUB">
<sequence name="Start">
<operation name="Human Interaction Event">
<participant name="HumanInteractionEvent"/>
<output message="HumanInteractionEventTypeInputMessage">
        <assign to="DocumentType">Purchase Order</assign>
        <assign to="FromAccount">Joe Supplier</assign>
        <assign to="Operation">UPDATE</assign>
        <assign to="State">Approved</assign>
        <assign to="Status">Read</assign>
        <assign to="StorageArea">Inbox</assign>
        <assign to="SuspendAsWaiting">NO</assign>
        <assign to="SystemAccount">admin</assign>
        <assign to="TemplateName">ExamplePurchaseOrders</assign>
        <assign to="MinedDataId" from="string(/ProcessData/list/listItem/@objectId)"/>
        <assign to="." from="*"/>
        </output>
        <input message="inmsg">
                <assign to="." from="*"/>
        </input>
</operation>
</sequence>
</process>
```

MinedDataId specified in BPML.

**Note:** If the business process used to add the document to the database is in a wait state, an update or delete causes the business process to resume processing.

# Chapter 98. Human Interaction Query Service

The Human Interaction Query service searches Sterling Integrator for business process data requiring human interaction and returns the query results to the business process. The query results include the reference data for documents that were previously saved in the Sterling Integrator database by the Human Interaction Event service. The query results also include the information about the state of the business process that was used to save the documents in the database (for example, whether or not the business process is in a waiting state).

Typically, after the Human Interaction Query service has completed, the Human Interaction XForms service identifies the Web template used to display the results returned from the service. The Human Interaction XForms service then works in conjunction with the B2B HTTP Server adapter to display the results in a Web browser.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

The following table provides an overview of the Human Interaction Query service:

| System name | HumanInteractionQuery |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Web Extensions |
| Description | Searches the Sterling Integrator WEBX_MINED_DATA table for records that match the query parameters specified for this service. The results from the query are saved in XML format and returned to the business process as a document. |
| Business usage | A business process uses the Human Interaction Event service to save documents requiring approval in the database and then sends out an e-mail notification to the approver. The e-mail notification includes the URL with the Web template for displaying the documents. When the approver specifies the URL in the browser, the Web template invokes a business process that uses the Human Interaction Query service to retrieve all documents requiring approval from the database and uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the results in the Web browser for user selection. **Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*. |

| | |
|---|---|
| Usage example | A manager logs into the Sterling Integrator Web extensions and wants to view all items in her inbox requiring approval. The process might include the following steps: The manager specifies a URL in the Web browser that includes the name of the Web template used to display the list of items. The Web template invokes a business process that uses the Human Interaction Query service to retrieve items from the manager's inbox prior to being displayed. The Web template is displayed with the list of items requiring approval by the manager. The manager selects and approves or rejects an item from the list. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Designed to work with:<br>• B2B Lookup System service<br>• Human Interaction Document Loader service<br>• Human Interaction XForms service |
| Application requirements | None |
| Initiates business processes? | None |
| Invocation | Runs as part of a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• WorkFlowContext.SUCCESS: Success, with no errors. There still might not be a result set, but no errors encountered by service.<br>  For example, the service may not find rows in the WEBX_MINED_DATA table meeting the query criteria and therefore no results are returned to the business process.<br>• WorkFlowContext.ERROR IllegalArgumentException: Caught IllegalArgumentException in QueryService.processData()<br>• WorkFlowContext.ERROR IllegalStateException: Caught IllegalStateException in QueryService.processData()<br>• WorkFlowContext.ERROR TransformerException: Caught TransformerException in QueryService.processData() |
| Restrictions | None |
| Persistence level | None |
| Testing considerations | None |

## Requirements

Using this service requires:
- Enabling the Human Interaction Event service configuration to store documents requiring Human Interaction in the Sterling Integrator database
- Enabling the Human Interaction XForms service configuration to retrieve items requiring Human Interaction from the Sterling Integrator database

- Creating a valid, deployed Web template for displaying data in a Web browser

## How the Human Interaction Query Service Works

Use the Human Interaction Query service to search the Sterling Integrator database for business process data that requires human interaction. The query results returned from this service include the reference data for documents that were previously saved in the Sterling Integrator database by the Human Interaction Event service.

## Business Scenario

Your company receives purchase orders in XML format from a trading partner. All purchase orders exceeding $1,000.00 need to be approved. You want to view a list of all purchase orders requiring approval.

## Business Solution Example

The approach used to solve this business scenario includes creating a business process that includes:
- A Human Interaction Query service to retrieve the list of purchase orders requiring approval
- A Human Interaction XForms service to receive the list of purchase orders from the Human Interaction Query service and identify the Web template used to display the list.
- A B2B HTTP Server adapter for displaying the list in a Web browser.

  **CAUTION:**
  **Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

## Business Solution Example Dependencies

The following dependencies must be met to run the example in this section and display the results in a Web browser:
- Run a business process that accepts purchase orders as input and uses the Human Interaction Event service to store the purchase orders in the Sterling Integrator database and mark them as requiring approval. The Human Interaction Event service must also be configured to invoke a business process that constructs and sends to the approver an e-mail notification containing the URL for displaying the list of purchase orders. See the business scenario and business solution example for the B2B Lookup System service for an example business process.
- Create a configuration of the HTTP Respond service that specifies the URI used to invoke (from the Web browser) the business process created in this example.
- Deploy a Web template in Sterling Integrator for displaying the list of purchase orders in the Web browser.

This business solution example focuses only on the Human Interaction Query service functionality.

## GPM Example

The following GPM example shows a solution for retrieving the list of purchase orders requiring approval to display in a Web browser.
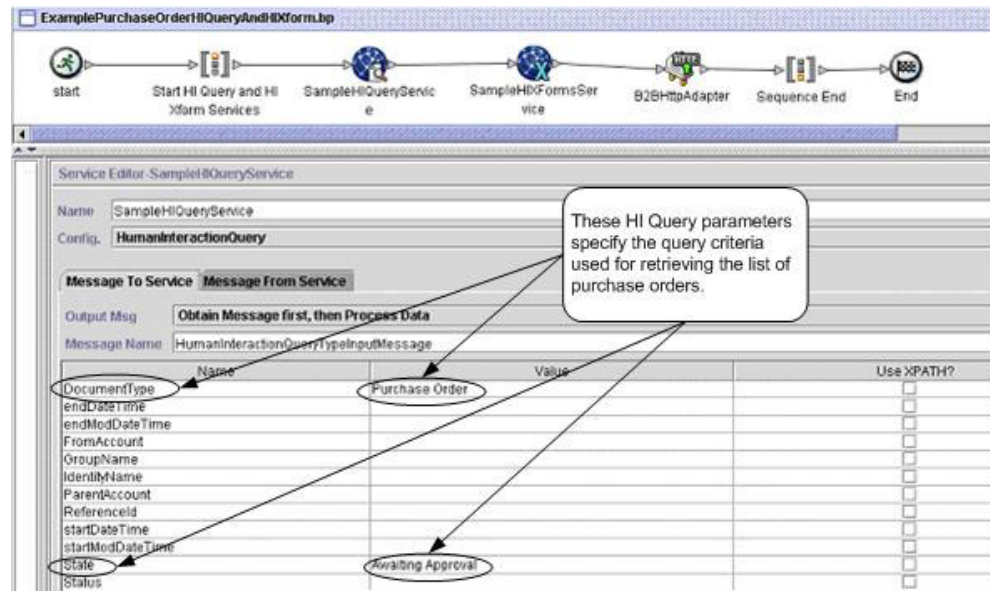


After the Human Interaction Query service completes, a list of purchase orders that are waiting for approval are returned to the business process in a document named MinedDataQueryResults. The returned document for this example looks like the following, returning reference data for the two purchase orders meeting the query criteria:

```
<?xml version="1.0" encoding="utf-8"?>
<list xmlns="">
<listItem objectId="1446e4:fb36e3f660:1c5f">
<referenceId>P01234</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner1</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>belushi:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/>
<fromMinedDataId>1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate><
marked/>
</listItem>
<listItem objectId="1446e4:fb36e3f660:1c67">
<referenceId>P04567</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner2</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>belushi:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/>
<fromMinedDataId>1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate>
<marked/>
</listItem>
</list>
```

Start of reference data for first purchase order

Start of reference data for second purchase order

**Note:** The Human Interaction Document Loader service can use the value of the objectId attribute on the listItem element to retrieve the actual purchase order document associated with the reference data.

The MinedDataQueryResults document is then sent to the Human Interaction XForms service and the B2B HTTP Server adapter for display in the Web browser.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

For example, you might have a Web template that displays the reference data in the following format:

## PURCHASE ORDER REQUESTS

| Order Number | Buyer | Request Date | State |
|---|---|---|---|
| PO1234 | TradingPartner1 | 03/20/2004 | Awaiting Approval |
| PO4567 | TradingPartner2 | 03/21/2004 | Awaiting Approval |

## Business Process Modeling Language (BPML) Example

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

```
<process name="ExamplePurchaseOrderHIQueryAndHIXform">
<sequence name="Start HI Query and HI Xform Services">
<operation name="SampleHIQueryService">
<participant name="HumanInteractionQuery"/>
<output message="HumanInteractionQueryTypeInputMessage">
 <assign to="DocumentType">Purchase Order</assign>
 <assign to="State">Awaiting Approval</assign>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
<operation name="SampleHIXFormsService">
<participant name="HumanInteractionXForms"/>
<output message="HumanInteractionXFormsTypeInputMessage">
 <assign to="TemplateName" from="DocToDOM(MinedDataQueryResults)/
listItem/templateName/text()"/>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
<operation name="HTTP Respond Service">
     <participant name="HttpRespond"/>
     <output message="HttpRespondServiceInputMessage">
       <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
       <assign to="." from="*"></assign>
     </input>
   </operation>
</sequence>
</process>
```

## Implementing the Human Interaction Query Service

To implement the Human Interaction Query service, complete the following tasks:

1. Activate your license for the Human Interaction Query service. See *Installing Sterling Integrator*.
2. Create a Human Interaction Query Service configuration. See *Managing Services and Adapters*.
3. Use the Human Interaction Query service in a business process.

## Human Interaction Query Service Business Process Usage

The following screen shows a graphical view of the GPM parameters for the Human Interaction Query service. There are no fields to be configured on the Message From Service tab.



The following example shows the corresponding BPML parameters for the Human Interaction Query service GPM parameters.

```
<process name="ExamplePurchaseOrderHIQuery">
<sequence name="Start HI Query Service">
<operation name="SampleHIQueryBP">
<participant name="HumanInteractionQuery"/>
<output message="HumanInteractionQueryTypeInputMessage">
 <assign to="DocumentType">Purchase Order</assign>
 <assign to="endDateTime">2004-02-25 00:00:00</assign>
 <assign to="endModDateTime">2004-02-25 00:00:00</assign>
 <assign to="FromAccount">SendingTradingPartnerSystemAccountID</assign>
 <assign to="GroupName">SendingTradingPartnerSecurityPermissionGroup</assign>
 <assign to="IdentityName">SendingTradingPartnerName</assign>
 <assign to="ParentAccount">ManagerOfTheDocumentSender</assign>
 <assign to="ReferenceId">PO1234</assign>
 <assign to="startDateTime">2004-02-24 00:00:00</assign>
 <assign to="startModDateTime">2004-02-24 00:00:00</assign>
 <assign to="State">Awaiting Approval</assign>
 <assign to="Status">Unread</assign>
 <assign  to="StorageArea">Inbox</assign>
 <assign to="SystemAccount" from="ReceiverCompanyUserAccount"/>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

The following table describes the fields used to configure the Human Interaction Query service in the GPM. The values specified for these parameters are the input to the Human Interaction Query service from the business process.

**Note:** If you do not specify any query criteria, no results are returned to the business process.

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| DocumentType | Type of data that requires human interaction. Valid value is any alphanumeric string, such as Purchase Order, Invoice, and Remittance Advice. |
| endDateTime | End date and time of the time period for the Human Interaction Query service to search Sterling Integrator for original business processes flagged for human interaction. The format for this field is *yyyy-mm-dd hh:mm:ss*, with a space between the *dd* and *hh*. The default is to return the reference data for all documents meeting the specified query criteria. |
| endModDateTime | End date and time of the time period for the Human Interaction Query service to search Sterling Integrator for modified business processes flagged for human interaction. The format for this field is *yyyy-mm-dd hh:mm:ss*, with a space between the *dd* and *hh*. The default is to return the reference data for all documents meeting the specified query criteria. |
| FromAccount | Trading partner system user account ID (as defined in Sterling Integrator) of the trading partner that sent the document. |
| GroupName | Name of the security permission group to which the trading partner that sent the document belongs. Use this parameter to search records pertaining to a specific security permission group used for the Human Interaction Event service. |
| IdentityName | Name of the trading partner that sent the document. Use this parameter to search records pertaining to a specific company. |
| ParentAccount | User account name for the manager who sent the document. This user account name is associated with the FromAccount information in this service. Use this parameter to search records pertaining to a specific manager. |
| ReferenceID | Document identifier. Valid value is any alphanumeric string. Examples include a purchase order number or an invoice number. |
| startDateTime | Start date and time for the Human Interaction Query service to search Sterling Integrator for original documents flagged for human interaction. The format for this field is *yyyy-mm-dd hh:mm:ss*, with a space between the *dd* and *hh*. The default is to return the reference data for all documents meeting the specified query criteria. |
| StartModDateTime | Start date and time for the Human Interaction Query service to search Sterling Integrator for modified documents flagged for human interaction. The format for this field is *yyyy-mm-dd hh:mm:ss*, with space between the *dd* and *hh*. The default is to return the reference data for all documents meeting the specified query criteria. |
| State | State of the document in the process. Valid value is any alphanumeric string, such as Approved, Pending, or Rejected. |

| Field | Description |
|---|---|
| Status | Status of the document. Valid value is any alphanumeric string, such as Read or Unread. |
| StorageArea | Virtual location to store the data waiting for the person to take action. Valid value is any alphanumeric string, such as Inbox, Outbox, and Drafts. |
| SystemAccount | Trading partner user account ID (as defined in Sterling Integrator) of the person who must interact with the business process. |

The following table describes additional fields used to configure the Human Interaction Query service. You can specify these fields using the Advanced Editor option in the GPM Service Editor or by using BPML.

| Field | Description |
|---|---|
| Sort Order | Used to specify the Mined Data column that is used to sort the results returned by the Human Interaction Query service. |
| State2 | An additional parameter that is only used by the *QuerySearch forms when the query includes more than just Active states, specifically when Active and Archive are the options. The Light Weight JDBC service can be use for queries that go beyond what the HI Query service can provide. |

## Output from Service to Business Process

The following table describes the output returned from the Human Interaction Query service to the business process:

| Returned Output | Description |
|---|---|
| MinedDataQueryRes ults/@SCIObjectID document | The name of the document, along with the document SCI Object ID, returned by the Human Interaction Query service. This document contains the list of document reference data meeting the query criteria specified by the Human Interaction Query service parameters. |

# Chapter 99. Human Interaction XForms Service

Use the Human Interaction XForms service to select the appropriate Web template for displaying business process data in a Web browser. The business process data could be documents and document reference data stored in the Sterling Integrator database or data returned from other services or adapters.

The Human Interaction XForms service:

- Generates the HTML for the Web page based on the Web template and posts the HTML to the B2B HTTP Server adapter for display in the Web browser.

  **Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

- Conducts data validation and calculations.

The following table provides an overview of the Human Interaction XForms service:

| System name | HumanInteractionXForms |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Web Extensions |
| Description | Directs Sterling Integrator to the appropriate Web template to use to display business process data. The Human Interaction XForms service displays the business process data in HTML format in a Web browser using a configured Web template. |
| Business usage | A business process uses the Human Interaction Event service to save documents requiring approval in the database and then sends out an e-mail notification to the approver. The e-mail notification includes the URL for displaying the documents. When the approver specifies the URL in the browser, a business process runs that uses the Human Interaction Query service to retrieve all documents requiring approval from the database and uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the results in the Web browser for user selection. **Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters.* |

| | |
|---|---|
| Usage example | You can use the Human Interaction XForms service in various ways. For example:<br><br>• The Human Interaction XForms service is preceded by a Human Interaction Query service that searches Sterling Integrator for business processes flagged for human interaction and saves the list in XML format. This list is passed to the Human Interaction XForms service, and the service identifies the Web template used to display the list.<br><br>• The Human Interaction XForms service is preceded by a Sterling Integrator service or adapter that produces XML as output and passes this output to the service for display in a Web browser. The Human Interaction XForms service identifies the Web template used to display the output. See *Usage Examples* for an example using this service to display data from the Lightweight JDBC adapter.<br><br>In each scenario, the Human Interaction XForms service is followed by the B2B HTTP Server adapter for displaying the information in the Web browser.<br>**Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Designed to work with:<br><br>• Human Interaction Query service<br><br>• Human Interaction Event service<br><br>• B2B HTTP Server adapter<br><br>• Human Interaction Document Loader service<br><br>**Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*. |
| 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: |
|---|---|
| | • WorkFlowContext.SUCCESS: Success, with no errors. |
| | • WorkFlowContext.ERROR IllegalArgumentException: Caught IllegalArgumentException in XFormsService.processData() |
| | • WorkFlowContext.ERROR IllegalStateException: Caught IllegalStateException in XFormsService.processData() |
| | • WorkFlowContext.ERROR SQLException: Caught SQLException in XFormsService.processData() |
| Restrictions | None |
| Persistence level | None |
| Testing considerations | None |

## Requirements

Using this service requires:
• A valid, deployed Web template
• An enabled B2B HTTP Server adapter configuration

> **Note:** The B2B HTTP Server adapter facilitates communications using the HTTP protocol. The B2B HTTP Server adapter must follow the Human Interaction XForms service for the document to be displayed in the Web browser.
> **CAUTION:**
> **Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters*.

## How the Human Interaction XForms Service Works

Use the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display business process data in a Web browser. The following sections describe a business scenario and a sample solution using the Human Interaction XForms service.

Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See *Retiring and Removed Services and Adapters*.

## Business Scenario

Your company receives purchase orders in XML format from a trading partner. All purchase orders exceeding $1,000.00 need to be approved. You want to view a list of all purchase orders requiring approval in the Web browser.

The approach used to solve this business scenario includes creating a business process that includes:
• A Human Interaction Query service to retrieve the list of purchase orders requiring approval
• A Human Interaction XForms service to:

– Receive the list of purchase orders from the Human Interaction Query service. Identify the Web template used to display the list.

– Forward the HTML from the Web template to the B2B HTTP Server adapter for displaying in a Web browser.

**CAUTION:**

**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

The following dependencies must be met to run the example in this section and display the results in a Web browser:

- Run a business process that accepts purchase orders as input and uses the Human Interaction Event service to store the purchase orders in the Sterling Integrator database and mark them as requiring approval. The Human Interaction Event service must also be configured to invoke a business process that constructs and sends an e-mail notification containing the URL for displaying the list of purchase orders to the approver. See the business scenario and business solution example for the Human Interaction Event service for an example business process.

- Create a configuration of the HTTP Respond service that specifies the URI used to invoke the business process created in this example from the Web browser.

- Deploy a Web template in Sterling Integrator for displaying the list of purchase orders in the Web browser.

This business solution example focuses only on the Human Interaction XForms service functionality.

## GPM Example

The following example shows a solution to the business scenario using the GPM.



The Human Interaction Event service (not shown) generated an e-mail notification to the approver after the purchase orders were added to the database. The approver uses this URL to list the purchase orders requiring approval in the Web

browser. For example, http://*siHostName:siport*/webx/bp/*businessProcessName*, where *businessProcessName* is the name of the business process to invoke.

The business process that runs uses the Human Interaction Query service to retrieve the list of purchase orders from the database and uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the list.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

After the Human Interaction Query service completes, a list of purchase orders that are waiting for approval are returned to the business process in a document named MinedDataQueryResults. The returned document for this example looks like the following, returning reference data for the two purchase orders:

```
<?xml version="1.0" encoding="utf-8"?>
<list xmlns="">
<listItem objectId="1446e4:fb36e3f660:1c5f">
<referenceId>PO1234</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner1</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>server1:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/><fromMinedDataId>
1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate>
<marked/>
</listItem>
<listItem objectId="1446e4:fb36e3f660:1c67">
<referenceId>PO4567</referenceId>
<operation>ADD</operation>
<storageSpace>Inbox</storageSpace>
<documentType>Purchase Order</documentType>
<systemAccount>TradingPartner2</systemAccount>
<templateName>ExamplePurchaseOrders</templateName>
<date>2004-03-12T18:02:49.000Z</date>
<waiting>NO</waiting>
<wfcId>server1:1446e4:fb36e3f660:1c5c</wfcId>
<workflowId>3013</workflowId>
<fromAccount/>
<fromMinedDataId>1446e4:fb36e3f660:1c5f</fromMinedDataId>
<identityName/>
<groupName/>
<parentAccountId/>
<state>Awaiting Approval</state>
<status/>
<timeout/>
<modifiedDate>2004-03-12T18:02:49.000Z</modifiedDate>
<marked/>
</listItem>
</list>
```

Web template used for displaying the data

The MinedDataQueryResults document is then sent to the Human Interaction XForms service, which identifies the Web template used for displaying the list of purchase orders and returns the HTML to the business process. The Human Interaction XForms service forwards the HTML to the B2B HTTP Server adapter for display in the Web browser.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

For example, you might have a Web template that displays the reference data in the following HTML format:

## PURCHASE ORDER REQUESTS

| Order Number | Buyer | Request Date | State |
|---|---|---|---|
| PO1234 | TradingPartner1 | 03/20/2004 | Awaiting Approval |
| PO4567 | TradingPartner2 | 03/21/2004 | Awaiting Approval |

The approver can select the purchase order number from the list to display the purchase order in the Web template and then approve or reject the order. See Human Interaction Document Loader service for information about retrieving a specific document from the database for display purposes.

The approver could alternatively log in to the Sterling Integrator Web Suite application to view (in their inbox) a list of documents requiring approval.

## Business Process Modeling Language (BPML) Example

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

```
<process name="ExamplePurchaseOrderHIQueryAndHIXform">
<sequence name="Start HI Query and HI Xform Services">
<operation name="SampleHIQueryService">
<participant name="HumanInteractionQuery"/>
<output message="HumanInteractionQueryTypeInputMessage">
 <assign to="DocumentType">Purchase Order</assign>
 <assign to="State">Awaiting Approval</assign>
 <assign to="SystemAccount" from="system-account-user-id/text()"/>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
<operation name="SampleHIXFormsService">
<participant name="HumanInteractionXForms"/>
<output message="HumanInteractionXFormsTypeInputMessage">
 <assign to="TemplateName" from="DocToDOM(MinedDataQueryResults)/
listItem/templateName/text()"/>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
<operation name="HTTP Respond Service">
     <participant name="HttpRespond"/>
     <output message="HttpRespondServiceInputMessage">
       <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
       <assign to="." from="*"></assign>
     </input>
   </operation>
</sequence>
</process>
```

See *Usage Examples* for an additional example of using the Human Interaction XForms service.

## Implementing the Human Interaction XForms Service

To implement the Human Interaction XForms service, complete the following tasks:

1. Activate your license for the Human Interaction XForms service. See *Installing Sterling Integrator*.
2. Create a Human Interaction XForms service configuration. See *Managing Services and Adapters*.
3. Use the Human Interaction XForms service in a business process.

## Human Interaction XForms Service Business Process Usage

The following screen shows a graphical view of the GPM parameters for the Human Interaction XForms service. There are no fields to be configured on the Message From Service tab.



The following example shows the corresponding BPML parameters for the Human Interaction XForms service GPM parameters.

```
<process name="ExampleHIXFormsGPMParms">
<sequence name="Start">
<operation name="SampleHIXFormsService">
<participant name="HumanInteractionXForms"/>
<output message="HumanInteractionXFormsTypeInputMessage">
 <assign to="ArchiveFlag">1</assign>
 <assign to="ArchiveHours">24</assign>
 <assign to="MergeDataOnResume">NO</assign>
 <assign to="ServletResponseMode">NO</assign>
 <assign to="TemplateName">WebTemplateName</assign>
 <assign to="." from="*"/>
</output>
<input message="inmsg">
 <assign to="." from="*"/>
</input>
</operation>
</sequence>
</process>
```

The following table describes the fields used to configure the Human Interaction XForms service in the GPM. The values specified for these parameters are the input to the Human Interaction XForms service from the business process.

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| ArchiveHours | The number of hours the mined data record should remain in the Web Extension table before it is archived or purged. Valid value is a positive integer. Optional. If blank, the time span from the business process is used, if it has been specified. If the time span from the business process has not been specified, the system default is used. |
| ArchiveFlag | The archive method that should be used. Optional. Valid values:<br><br>• 1 – Archives the data in the Web Extension table.<br>• 2 – Deletes the data from the Web Extension table.<br>• blank – The archive method from the business process is used. |
| MergeDataOnResume | Merges any changes made to the Web template with the existing instance data when the business process is taken out of the waiting state. Valid values:<br><br>• Yes – Causes the documents and instance data to be merged when the business process is resumed.<br>• No – Overlays the business process being resumed (taken out of a waiting state) with the changed documents and instance data. Default. |
| ServletResponseMode | Determines the operation of the Human Interaction XForms service. Valid values:<br><br>• Yes – Recognizes that the business process started by a Business Process XLink Resolver. The Human Interaction XForms service disregards all other configured parameters and immediately sends the primary document back to the XForms servlet for processing.<br>• No – Parameter has no affect. The service works as configured. Default.<br><br>**Note:** The Business Process XLink Resolver is a Java class that starts business processes and is used in place of the HTTP Servlet adapter for Web Extensions running on Sterling Integrator. In this case, business processes start directly by a call to the workflow class. |
| TemplateName | Name of the Web template to use for viewing the data. Select the template name from the list of deployed templates, or dynamically select the template name from the query results returned by the Human Interaction Query service. |

## Usage Examples

The example in this section illustrates a business process that uses the Human Interaction XForms service in conjunction with the B2B HTTP Server adapter to display the results returned from a Lightweight JDBC adapter. The Lightweight JDBC adapter is used to query the Sterling Integrator database for customer

records and return the results as a primary document to the business process. The primary document is passed to the Human Interaction XForms service, which identifies the Web template and generates the HTML for displaying the customer information. The Human Interaction XForms service then posts the document to the B2B HTTP Server adapter for display in the Web browser.

**CAUTION:**
**Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. See** *Retiring and Removed Services and Adapters***.**

This business process started by specifying a URL in the Web browser (for example, http://*siHostName*:*siPort*/webx/bp/ExampleCustomerList), which is accomplished with the Lightweight JDBC adapter.

**Note:** This example assumes that there is a valid, deployed Web template in Sterling Integrator for displaying the list of customers in the Web browser and that there is URI mapping defined in the B2B HTTP Server adapter configuration for invoking the business process from an HTTP request.

The following example illustrates this scenario using the GPM.
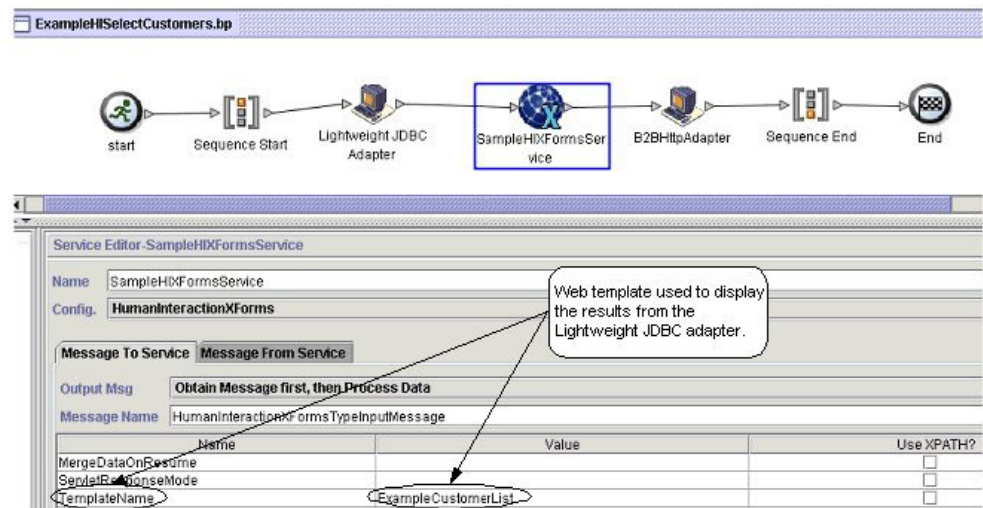


The Lightweight JDBC adapter returns to the business process the following customer list as the primary document. This document is input to the Human Interaction XForms service.

```
<?xml version='1.0'
encoding='UTF-8'?>
<CustomerList>
<RowsReturnedFromDatabase>
<customer_id>1234</customer_id>
<customer_name>John Doe</customer_name>
<customer_address>address1</customer_address>
<customer_phone>xxxx-xxx-xxxx</customer_phone>
</RowsReturnedFromDatabase>
<RowsReturnedFromDatabase>
<customer_id>5678</customer_id>
<customer_name>Jane Doe</customer_name>
```

```
<customer_address>address1</customer_address>
<customer_phone> xxxx-xxx-xxxx</customer_phone>
</RowsReturnedFromDatabase>
</CustomerList>
```

The user specifies the URL in the Web browser for example, http://
*siHostName*:*siPort*/webx/bp/ExampleCustomerList), which invokes the
ExampleHISelectCustomers business process and displays the list of customers in
the Web browser. For example, you might have a Web template that displays the
list of customers in the following HTML format:

**CUSTOMER LIST**

| Customer ID | Name | Address | Phone |
|---|---|---|---|
| 1234 | John Doe | address1 | xxx-xxx-xxxx |
| 4567 | Jane Doe | address2 | xxx-xxx-xxxx |

The following example illustrates the business process using BPML:

```
<process name="ExampleHISelectCustomers">
<sequence>
<operation name="Lightweight JDBC Adapter">
     <participant name="ExampleLWJDBCBusinessProcess"/>
     <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="pool">mysqlTrainingPool</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="result_name">CustomerList</assign>
        <assign to="row_name">RowsReturnedFromDatabase</assign>
        <assign to="sql">SELECT * FROM Customer</assign>
        <assign to="." from="*"/>
     </output>
     <input message="inmsg">
        <assign to="Document" from="PrimaryDocument/@SCIObjectID"/>
     </input>
</operation>
<operation name="SampleHIXFormsService">
     <participant name="HumanInteractionXForms"/>
     <output message="HumanInteractionXFormsTypeInputMessage">
         <assign to="TemplateName">ExampleCustomerList</assign>
        <assign to="." from="*"/>
     </output>
     <input message="inmsg">
         <assign to="." from="*"/>
     </input>
</operation>
<operation name="HTTP Respond Service">
     <participant name="HttpRespond"/>
     <output message="HttpRespondServiceInputMessage">
       <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
       <assign to="." from="*"></assign>
     </input>
   </operation>
</sequence>
</process>
```

# Chapter 100. HTTP Client Adapter

The following table provides an overview of the HTTP Client adapter:

| System name | HTTP Client Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | None |
| Description | Sends HTTP requests to trading partners using a Perimeter server. The HTTP Client adapter replaces the B2B HTTP Client adapter and the HTTP Send adapter, which are being retired. The HTTP Client adapter and its related services provide all the functionality of the former two adapters plus the following enhancements:<br><br>• Use Perimeter services<br>• Highly scalable (>150 concurrent transfers)<br>• Supports large files up to 2 GB<br>• Supports HTTP 1.1 |
| Business usage | Use this adapter to send documents to a trading partner using HTTP. |
| Usage example | The Sterling Integrator business process must send a message to a trading partner and the trading partner's profile specifies HTTP as the transport protocol. The business process passes the document as well as information about the trading partner to the HTTP Client adapter. Using the information provided in the business process, the HTTP Client adapter connects to the trading partner and transfers the document, enforcing any security specifications from the business process. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br><br>• HTTP Client Begin Session Service<br>• HTTP Client End Session Service<br>• HTTP Client GET Service<br>• HTTP Client Method Service<br>• HTTP Client POST Service |
| Application requirements | An HTTP server is required 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. This Perimeter server is typically installed in a DMZ environment, separated from Sterling Integrator by a firewall. |
| Initiates business processes? | No |
| Invocation | The HTTP Client adapter is invoked from one of the HTTP Client services used within a business process. |
| Business process context considerations | Business processes using HTTP client services should not be marked as Auto Resume. These services require an established session which will no longer exist after a restart. |

| Returned status values | Returned status values: |
|---|---|
| | • 100s – Informational parameters and more status parameters will follow. |
| | • 200s – Request was OK. |
| | • 300s – Request failed and more status codes will follow. |
| | • 400s – Client Request Failed. This is a client error. |
| | • 500s – Server failed to handle a valid request. This is a server error. |
| | **Note:** For information on setting the SOAP fault status to return 500 HTTP error status, see *SOAP Outbound service*. |
| Restrictions | All work bound within an HTTP Client Begin Session service and an HTTP Client End Session service must be within the same business process. |
| Persistence level | N/A |
| Testing considerations | To test this adapter, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the *Business Process Example*. Debug information related to this adapter can be found in the HTTP Client Adapter and Services log. |

## Implementing the HTTP Client Adapter

To implement the HTTP Client adapter, complete the following tasks:

1. Create an HTTP Client adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client adapter. For information, see *Configuring the HTTP Client Adapter*.

## Configuring the HTTP Client Adapter

To configure the HTTP Client adapter, you must specify field settings in 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:<br><br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |
| Perimeter Server Name | List of Perimeter servers, including local-mode Perimeter servers. Required. |
| Local port range | Ranges or lists of local ports to restrict the adapter to. For example, 123-456 or 123,124,999. |
| Global Proxy Server | Available Global Proxy Servers. Select a global proxy server to enable all adapters to share a common proxy configuration. You can configure changes to all the proxy servers globally instead of modifying each adapter instance. For additional information on global proxy server, see Global Proxy Server Setting.<br>**Note:** If you have enabled both Use proxy server and Global Proxy Server parameters with specific proxy settings, the value entered for the Use proxy server parameter overrides the setting in the Global Proxy Server. |
| Number of connection retries | Number of times the HTTP Client adapter will try to connect to the server. Required. Valid value is any integer between 0 and 50. Default is 3. |
| Delay between retries (second) | Number of seconds the HTTP Client adapter will wait between retry attempts. Required. Valid value is any integer between 1 and 7200. Default is 20. |
| Response timeout (second) | Number of seconds it should take for the server to respond. Required. Valid value is any integer between 1 and 999999. Default is 60. |
| Use proxy server | Whether to use an HTTP proxy server. Required. Valid values:<br><br>• Yes – A proxy server will be used<br>• No – No proxy server will be used |
| Proxy hostname | IP address or host name of the proxy server to use. Required if Use proxy server is set to Yes. This parameter can be overridden by a Trading Partner profile setting. |
| Proxy port | Port number of the proxy server. Required if Use proxy server is set to Yes. This parameter can be overridden by a Trading Partner profile setting. |
| Proxy retry attempts | Number of times the HTTP Client adapter will try to connect to the proxy server. Required if Use proxy server is set to Yes. Valid values: any integer between 0 and 50. Default is 3. This parameter can be overridden by a Trading Partner profile setting. |

| Field | Description |
|---|---|
| Use Proxy Basic Authentication | Whether the proxy server requires basic authentication. Required if Use proxy server is set to Yes. Valid values:<br><br>• Yes – Proxy basic authentication will be used<br>• No – Proxy basic authentication will not be used |
| Proxy username | User name to use with the proxy server. Required if Use Proxy Basic Authentication is set to Yes. |
| Proxy password | Password for the Proxy username. Required if User Proxy Basic Authentication is set to Yes. |

## Business Process Example

The following example business process illustrates the use of the HTTP Client services, which work through the HTTP Client adapter:
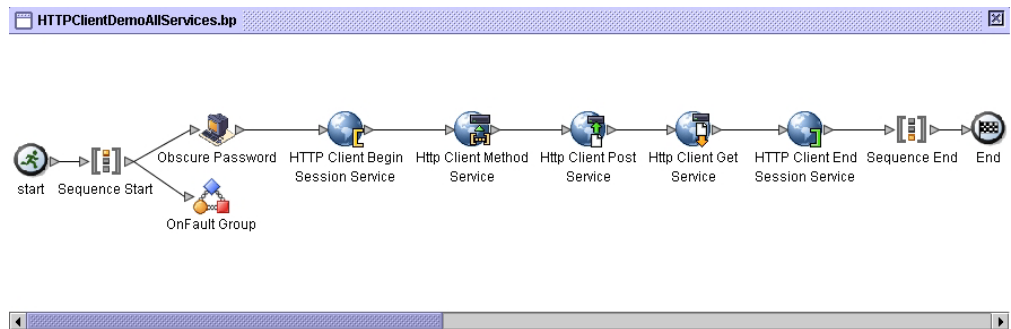


The associated BPML code is shown below:

```
<process name="HTTPClientDemoAllServices">
 <sequence>
   <!-- Get obscured password -->
   <operation name="Obscure Password">
     <participant name="HTTPClientObscureParameter"/>
     <output message="outmsg">
       <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
       <assign to="HTTPClientObscureResults" from="*"></assign>
     </input>
   </operation>
   <!-- HTTP Client Begin Session service-->
   <!-- Create connection to specified host and port -->
   <operation name="HTTP Client Begin Session Service">
     <participant name="HTTPClientBeginSession"/>
     <output message="HTTPClientBeginSessionServiceTypeInputMessage">
       <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
       <assign to="RemoteHost">httpserver</assign>
       <assign to="RemotePort">12345</assign>
       <!-- If server requires Basic authentication -->
       <assign to="RemoteUserId">userid</assign>
       <assign to="RemotePasswd"
from="HTTPClientObscureResults/admin/text()"></assign>
         RemoteUserId and RemotePasswd are not needed if using a
         trading partner profile. The profile will supply this information.
       <!-- To use a Trading Partner profile -->
       <assign to="ProfileId">someExistingProfileId</assign>
       <!-- To override HTTPClientAdapter configuration settings -->
       <assign to="ConnectionRetries">10</assign>
```

```
              <assign to="RetryDelay">1</assign>
               <!-- If server requires SSL authentication -->
              <assign to="SSL">Must</assign>
              <assign to="CipherStrength">Strong</assign>
              <assign to="CACertificateId">SomeHttp-Id</assign>
              <assign to="SystemCertificateId">httpsampleclientcert1-Id</assign>
              <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
              <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
          </input>
</operation>
      <!-- HTTP Client Method Service -->
    <operation name="HTTP Client Method Service">
      <participant name="HTTPClientMethod"/>
      <output message="HTTPClientMethodServiceTypeInputMessage">
        <assign to="SessionToken"  from="HTTPClientBeginSessionServiceResults/
            SessionToken/text()"></assign>
        <assign to="MethodType">HEAD</assign>
        <assign to="URI">/hello</assign>
        <assign to="RawResponse">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientMethodServiceResults" from="*" append="true"/>
      </input>
    </operation>
    <!-- HTTP Client POST Service -->
    <operation name="HTTP Client Post Service">
      <participant name="HTTPClientPost"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
            SessionToken/text()"></assign>
        <assign to="URI">/hello</assign>
        <assign to="RawResponse">false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientPostServiceResults" from="*" append="true"/>
      </input>
    </operation>
    <!-- HTTP Client GET Service -->
    <operation name="HTTP Client Get Service">
      <participant name="HTTPClientGet"/>
      <output message="HTTPClientGetServiceTypeInputMessage">
        <assign to="SessionToken"  from="HTTPClientBeginSessionServiceResults/
            SessionToken/text()"></assign>
        <assign to="URI">/someURI/</assign>
        <assign to="ResponseTimeout">120</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientGetServiceResults" from="*" append="true"/>
      </input>
    </operation>
     <!-- HTTP Client End Session Service -->
     <!-- Ends session specified by SessionToken -->
    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
            SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientEndSessionServiceResults" from="*" append="true"/>
      </input>
```

```
        </operation>
      <!-- Provides error handling -->
      <onFault>
        <sequence name="End Session">
        <operation name="HTTP Client End Session Service">
          <participant name="HTTPClientEndSession"/>
          <output message="HTTPClientEndSessionServiceTypeInputMessage">
            <assign to="SessionToken"  from="HTTPClientBeginSessionServiceResults/
                    SessionToken/text()"></assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
          </input>
        </operation>
        </sequence>
       </onFault>
      </sequence>
    </process>
```

# Chapter 101. HTTP Client Begin Session Service

The following table provides an overview of the HTTP Client Begin Session service:

| System name | HTTP Client Begin Session Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > HTTP Client |
| Description | The HTTP Client Begin Session service is used to start an HTTP session with an external trading partner for the purpose of exchanging business documents. This service works through an instance of the HTTP Client adapter. |
| Business usage | Use this service to establish a session with a trading partner HTTP server. |
| Usage examples | A business process is executed that translates a document that must be sent to a trading partner. After the translation, Sterling Integrator looks up information about how to transport data to the trading partner identified in the trading partner profile. The trading partner profile specifies HTTP as the transport protocol. Sterling Integrator then uses the HTTP Client Begin Session service to establish a session with the trading partner's HTTP server. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services: <br> • HTTP Client adapter <br> • HTTP Client End Session service <br> • HTTP Client GET service <br> • HTTP Client Method service <br> • HTTP Client POST service <br><br> To mask the values associated with the remote password parameter, use the Obscure Data - Process Data Values service in conjunction with the HTTP Begin Session service. This service is presented in the GPM as Obscure Parameter on the All Services stencil. |
| Application requirements | An HTTP Server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | The HTTP Client Begin Session service allows you to specify a remote password. In order for this password to be obscured in process data for the business process, you should also use the Obscure Data - Process Data Values service within the same business process. The Obscure Data - Process Data Values service may be used to mask the values associated with parameters. |
| Returned status values | Returned status values: <br> • 0 –Success <br> • 1 – Error |

| Restrictions | N/A |
|---|---|
| Persistence level | System Default |
| Testing considerations | To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the HTTP Client adapter. Debug information for this service can be found in the HTTP Client adapter and services log files. |

## Implementing the HTTP Client Begin Session Service

To implement the HTTP Client Begin Session service, complete the following tasks:

1. Create an HTTP Client Begin Session service configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client Begin Session service. For information, see *Configuring the HTTP Client Begin Session Service*.
3. Use the HTTP Client Begin Session service in a business process.

## Configuring the HTTP Client Begin Session Service

You can set the following values in the trading partner profile and specify it in the ProfileId field, or you can set these values in an instance of the service to only apply for that instance. If specified in the HTTP Client Begin Session service, the following values override those in the HTTP trading partner profile:

- CACertificateId
- CipherStrength
- ConnectionRetries
- RemoteHost
- RemotePasswd
- RemotePort
- RemoteUserId
- SSL
- SystemCertificateId
- RetryDelay

To configure the HTTP Client Begin Session service, you must specify settings for the following fields in the GPM:

| Field | Description |
|---|---|
| Name | Name this service will have in Sterling Integrator |
| Description | Description of service |

| Field | Description |
|---|---|
| Select a Group | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |
| Config | Name of the service configuration. |
| CACertificateId | Select from the list of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID. Required if SSL is set to Must. Check an SSL certificate into the application to make it available in this list. |
| CipherStrength | The level of encryption to apply to the data that flows through the socket connection. Optional. Valid values are:<br><br>• ALL – WEAK or STRONG is accepted<br>• WEAK – 40 bit encryption is required<br>• STRONG – 40 bit or higher encryption is required (default) |
| ConnectionRetries | The number of times the service will try to connect to the Trading Partner Systems. Connection retries occur only with TCP/IP related issues. Optional. Valid value is any numeric value.<br>**Note:** The value entered for this parameter overrides the **Number of connection retries** setting in the HTTP Client adapter configuration. |
| DelayWaitingOnIO | The amount of time (in seconds) that a business process using the Http Client Begin Service adapter waits before going to WAITING_ON_IO state and frees up the engine resources for other processes. Optional. Valid value is an integer. If you specify a positive integer, the parameter specifies the number of seconds the business process has to wait for a response from the HTTP server before going to WAITING_ON_IO state. If you specify a negative integer, the business process waits for the response from the HTTP server to complete. The business process does not go to WAITING_ON_IO state. If you specify 0, the business process goes to WAITING_ON_IO state after sending a request to the HTTP server. If you specify a value lesser than -1, the parameter value is set to 0 (default value).<br>**Note:** The value you specify in the httpclient.properties file for the defaultDelayWaitingOnIO property overrides the setting you specified in the GPM. |
| HTTPClientAdapter | Select the HTTP Client adapter for this service to use when beginning a session with an HTTP server. Required. |

| Field | Description |
|---|---|
| ProfileId | Trading partner profile identification. Optional. Valid value is any valid profile ID. |
| RemoteHost | External Trading Partner host system (HTTP server IP Address or DNS name. Required. Use any valid IP Address or DNS name. |
| RemotePasswd | HTTP remote login password. Optional.<br>**Note:** For the password to be masked in process data, the Obscure Data - Process Data Values service must also be used in the same business process. The name used to store the password must be the same as the specified RemoteUserId. |
| RemotePort | External Trading Partner port number. Required. |
| RemoteUserId | HTTP remote login username. Optional. |
| RetryDelay | Number of seconds the adapter will wait before retrying. Optional. Valid value is any numeric value.<br>**Note:** The value entered for this parameter overrides the **Delay between retries** setting in the HTTP Client adapter configuration. |
| SSL | Determines SSL socket negotiation. Optional. Valid values are:<br>• Must - SSL socket negotiation is enabled.<br>• None -Connect will not use SSL. Default. |
| SystemCertificateId | Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. This certificate confirms the identity of the client to the server. Required if SSL is set to Must and the server requires client authentication. Obtain the certificate from your trading partner. Check it into Sterling Integrator from the Admin menu selecting **Trading Partner** > **Digital Certificates** > **System** to make it available in this list. |

## Output from Service to Business Process

The following table contains the parameters passed from the HTTP Client Begin Session service to the business process:

| Parameter | Description |
|---|---|
| SessionToken | Specifies the identifier for the session established between the HTTP Client adapter and an HTTP server. Required. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client Begin Session service:

| Field | Description |
|-------|-------------|
| CACertificateId | Drop-down menu that contains a list of trusted Certificate Authority public certificates. In process data, this parameter is displayed as an object ID. Required if SSL is set to Must. |
| CipherStrength | The level of encryption to apply to the data that flows through the socket connection. Valid values are:<br>• ALL – WEAK or STRONG is accepted<br>• WEAK – 40 bit encryption is required<br>• STRONG – 40 bit or higher encryption is required |
| HTTPClientAdapter | Select the HTTP Client adapter for this service to use when beginning sessions with an HTTP server. Required. |
| ConnectionRetries | The number of times the service will try to connect to the Trading Partner Systems. Connection retries occur only with TCP/IP related issues. Optional. Valid value: any numeric value. Default is 1. |
| ProfileId | Trading partner profile identification. Optional. Valid value is any valid profile ID. |
| RemoteHost | External Trading Partner host system (HTTP server IP Address or DNS name. Required. Use any valid IP Address or DNS name. |
| RemotePasswd | HTTP remote login password. Optional.<br>**Note:** The password will be obscured using the Obscure service. |
| RemotePort | External Trading Partner port number. Required. |
| RemoteUserId | HTTP remote login username. Optional. |
| RetryDelay | Number of seconds the adapter will wait before retrying. Optional. Valid value is any numeric value. Default is 1. |
| SessionBeginTime | Specifies the date and time that the session started. Required. |
| SSL | The SSL flag that determines SSL socket negotiation. Valid values are:<br>• Must - SSL socket negotiation is enabled.<br>• None -Connect will not use SSL. Default. |
| SystemCertificateId | Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner Trusted Certificate Authority. Valid value is any alphanumeric string. |
| UsingRevealedPasswd | Indicates whether the password sent to the service is unobscured. Valid value is True or False. Default is False. Optional. |

## Business Process Example

The following example business process illustrates using the HTTP Client Begin Session service:

```
<process name="HTTPExample">
  <sequence>
<operation name="Obscure Password">
      <!-- insert obscured password into process data -->
      <participant name="HTTPClientObscureParameter"/>
      <output message="outmsg">
        <assign to="." from="*"></assign>
      </output>
```

```
                <input message="inmsg">
                  <assign to="HTTPClientObscureResults" from="*"></assign>
                </input>
            </operation>
        <operation name="HTTP Client Begin Session Service">
                <participant name="HTTPClientBeginSession"/>
                <output message="HTTPClientBeginSessionServiceTypeInputMessage">
                  <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
                  <assign to="RemoteHost">hostb</assign>
                  <assign to="RemotePort">26633</assign>
        <assign to="RemoteUserId">admin</assign>
        <!-- copy obscured password from process data to service -->
        <assign to="RemotePasswd" from="admin/text()"></assign>
        <assign to="SSL">Must</assign>
                  <assign to="CipherStrength">Strong</assign>
                  <assign to="CACertificateId">B2BHttp-Id</assign>
                  <assign to="SystemCertificateId">httptestclientcert1-Id</assign> -->
                  <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
                  <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
                </input>
            </operation>
```

The following example shows how to use of the revealObscured function when the user ID contains domain or special characters. First, create a name in the Obscure Service without any special characters and assign the appropriate password to it. In the following example, the name created in the Obscure Service is "htan" while the userid is sgp-htan\htan.

The parameter to the revealObscured() function is the node containing the obscured password. The function uses the node name as the key and the node value as the obscured password when unobscuring.

```
<operation name="Obscure HTTP client password">
        <participant name="HTTPClientObscureParameter"/>
        <output message="outmsg">
 <assign to="." from="*"/>
        </output>
        <input message="inmsg">
 <assign to="ObscureResult" from="*"/>
        </input>
</operation>
<operation name=" HTTP Client Begin Session Service ">
        <participant name="HTTPClientBeginSession"/>
        <output message="BeginSessionRequest">
            ............
          <assign to="RemoteUserId">sgp-htan\htan</assign>
          <assign to="UsingRevealedPasswd">true</assign>
          <assign to="RemotePasswd" from="revealObscured(ObscureResult/htan)"/>
          ..........
        </output>
        <input message="inmsg">
            <assign to=" HTTPClientBeginSessionServiceResults " from="*"/>
        </input>
</operation>
```

# Chapter 102. HTTP Client End Session Service

The following table provides an overview of the HTTP Client End Session service:

| | |
|---|---|
| System name | HTTP Client End Session Service |
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > HTTP Client |
| Description | This adapter is used to end an HTTP session with an external trading partner HTTP server. This service works through an instance of the HTTP Client adapter. |
| Business usage | A business user would use this service as the last functional activity in a business process that sends an HTTP request to a trading partner. This service can only be used if the HTTP Client Begin Session Service was used at the start of the business process. |
| Usage example | A Sterling Integrator business process is executed that translates a document that must be sent to a trading partner. After the translation, Sterling Integrator looks up information about how to transport data to the trading partner in the trading partner profile. The trading partner profile specifies HTTP as the transport protocol. Sterling Integrator then begins a session with the trading partner using the HTTP Client Adapter, sends the document, then ends the session using the HTTP Client End Session Service. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• HTTP Client adapter<br>• HTTP Client Begin Session service<br>• HTTP Client GET service<br>• HTTP Client Method service<br>• HTTP Client POST service |
| Application requirements | An HTTP server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 0 –Success<br>• 1 –Error |
| Restrictions | None |
| Persistence level | System Default |

| Testing considerations | To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the *HTTP Client adapter*. Debug information for this service can be found in the HTTP Client adapter and services log files. |
| --- | --- |

## Implementing the HTTP Client End Session Service

To implement the HTTP Client End Session service, complete the following tasks:

1. Create an HTTP Client End Session service configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Client End Session service. For information, see *Configuring the HTTP Client End Session Service*.
3. Use the HTTP Client End Session service in a business process.

## Configuring the HTTP Client End Session Service

To configure the HTTP Client End Session service, you must specify field settings in the GPM:

| Field | Description |
| --- | --- |
| Config | Name of the service configuration. |
| SessionToken | Specifies the identifier of the session to be ended. Required. |

## Output from Service to Business Process

The following table contains the parameters passed from the HTTP Client End Session service to the business process:

| Parameter | Description |
| --- | --- |
| SessionEndTime | Specifies the date and time that the session ended. Required. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client End Session service:

| Parameter | Description |
| --- | --- |
| SessionToken | Specifies the ID of the session to be ended. Required. |

## Business Process Example

The following example business process illustrates using the HTTP Client End Session service:

```
<process name="default">
 <sequence>
 [[Insert Begin session operation here]]
   <operation name="HTTP Client End Session Service">
```

```
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken">SessionToken</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

# Chapter 103. HTTP Client GET Service

The following table provides an overview of the HTTP Client GET service:

| System name | HTTP Client GET Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > HTTP Client |
| Description | The HTTP Client GET service sends HTTP GET requests to a trading partner's HTTP server via the perimeter server. This service works in conjunction with the HTTP Client Begin service and the HTTP Client End service, and through an instance of the HTTP Client adapter. |
| Business usage | A business user would use the HTTP Client GET service to retrieve documents from a trading partner's HTTP server. |
| Usage example | A Sterling Integrator business process is executed that must retrieve a specified file from the external trading partner. Sterling Integrator uses the HTTP Client GET Service, working through the HTTP Client Adapter, to retrieve the file from the trading partner's HTTP server. The data is passed to the business process. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported platforms for Sterling Integrator |
| Related services | Related services:<br>• HTTP Client adapter<br>• HTTP Client Begin Session service<br>• HTTP Client End Session service<br>• HTTP Client Method service<br>• HTTP Client POST service |
| Application requirements | An HTTP server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | The HTTP Client GET service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 100s – Informational parameters and more status parameters will follow.<br>• 200s – Request was OK.<br>• 300s – Request failed and more status codes will follow.<br>• 400s – Client Request Failed. This is a client error.<br>• 500s – Server failed to handle a valid request. This is a server error. |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the *HTTP Client adapter*. Debug information for this service can be found in the HTTP Client adapter and services log files. |

## Implementing the HTTP Client GET Service

To implement the HTTP Client GET service, complete the following tasks:

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

## Configuring the HTTP Client GET Service

To configure the HTTP Client GET service, you must specify field settings in the Graphical Process Modeler (GPM):

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| DocumentId | A document ID to use with this instance of the HTTP Client GET service. Optional. |
| RawResponse | Whether to include the HTTP Server response headers in the response document. Optional. Valid values:<br>• true – Both the HTTP headers and the entity body are copied to the body of the business process document.<br>• false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process.<br>Default is false. |
| ResponseTimeout | Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.<br>**Note:** The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration. |
| SessionToken | Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required. |
| ShowResponseCode | Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:<br>• true – Metadata will be included.<br>• false – Metadata will not be included. Default is false. |
| URI | Uniform Resource Indicator (URI) representing the HTTP server. Required. |

| Field | Description |
|---|---|
| Cookie | Identifies the cookie to use in cases where the previous GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Get Service uses the Cookie parameter and creates the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client GET service:

| Field | Description |
|---|---|
| DocumentId | A document ID to use with this instance of the HTTP Client GET service. Optional. |
| RawResponse | Whether to include the HTTP Server response headers in the response document. Optional. Valid values:<br><br>• true – Both the HTTP headers and the entity body are copied to the body of the business process document.<br><br>• false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process.<br><br>Default is false. |
| ResponseTimeout | Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid. **Note:** The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration. |
| SessionToken | Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required. |
| ShowResponseCode | Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:<br><br>• true – Metadata will be included.<br><br>• false – Metadata will not be included. Default is false. |
| URI | Uniform Resource Indicator (URI) representing the HTTP server. Required. |
| Cookie | Identifies the cookie to use in cases where the previous GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Get Service uses the Cookie parameter and creates the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional. |

## Business Process Examples

The following BPML example illustrates using commands supported by the HTTP Client GET service.

```
<process name="HTTPClientGETServiceExample">
  <sequence>
  [[Insert begin session operation here]]
    <operation name="HTTP Client GET Service">
    <participant name="HTTPClientGETService"/>
    <output message="HTTPClientGetServiceTypeInputMessage">
      <assign to="DocumentId">DocumentID</assign>
      <assign to="RawResponse">true</assign>
      <assign to="ResponseTimeout">60</assign>
      <assign to="SessionToken">SessionToken</assign>
      <assign to="ShowResponseCode">true</assign>
      <assign to="URI">URI</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
    </operation>
  [[Insert end session operation here]]
  </sequence>
</process>
```

The following business process provides an example of using the Cookie parameter:

```
<process name="HTTP_To_advancepcsrx">
  <!-- Loop Invariant. -->
  <rule name="haveMoreCookie1">
 <condition> string(counterCookie) &lt;= count(HTTPClientPostServiceResults/
    ServerResponse/Headers/Set-Cookie/node()) </condition>
  </rule>
  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="CACertificateId">sgmillenia:13582d:10682043f1d:-73bd</assign>
        <assign to="CipherStrength">all</assign>
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="SSL">Must</assign>
        <assign to="RemoteHost">webtransport.advancepcsrx.com</assign>
        <assign to="RemotePort">443</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>
    <!-- Use FSA to pick up the input file -->
    <operation name="Import Document Request">
      <participant name="TEST_FILE_SYSTEM_ADAPTER"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder" from="'/ais_local/share/sli'"></assign>
        <assign to="filter" from="'AdvancePCS_URI.txt'"></assign>
        <assign to="useSubFolders">false</assign>
        <assign to="bootstrap">false</assign>
        <assign to="deleteAfterCollect">false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="FileSystemOutputMessage">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <!-- Set document content type/subtype -->
    <operation name="SetContenType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
```

```
        <assign to="." from="*"></assign>
        <assign to="DocumentContentType">application</assign>
        <assign to="DocumentContentSubType">x-www-form-urlencoded</assign>
      </output>
      <input message="xin">
        <assign to="docInfo" from="*"></assign>
      </input>
    </operation>
            <!-- POST to URI /template/login to login the webpage -->
    <operation name="Http Client Post Service">
      <participant name="HTTPClientPost"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
              SessionToken/text()"></assign>
        <assign to="URI">/template/login</assign>
        <assign to="RawResponse">true</assign>
        <assign to="ResponseTimeout">120</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientPostServiceResults" from="*" append="true"></assign>
      </input>
    </operation>
    <!-- concat the cookie from the POST Response -->
    <assign to="counterCookie">1</assign>
    <assign to="Cookie" from="''"/>
    <choice>
<select>
 <case ref="haveMoreCookie1" activity="AppendCookie1"/>
 </select>
    <sequence name="AppendCookie1">
 <assign to="Cookie" from="concat(string(Cookie), substring-before
  (//HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie
  [number(//counterCookie)],';'), '; ')"/>
 <assign to="counterCookie" from="number(counterCookie) + 1"/>
  <choice>
<select>
    <case ref="haveMoreCookie1" activity="Repeat1"/>
  </select>
<repeat name="Repeat1" ref="AppendCookie1"/>
</choice>
 </sequence>
    </choice>
    <!-- release Primary Document before GET -->
    <operation>
    <participant name="ReleaseService"/>
    <output message="releaseRequest">
      <assign to="TARGET" from="'PrimaryDocument'"/>
    </output>
    <input message="releaseResponse"/>
    </operation>
    <!-- GET URI / to reach the final page -->
    <operation name="Http Client Get Service">
      <participant name="HTTPClientGet"/>
      <output message="HTTPClientGetServiceTypeInputMessage">
        <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
              SessionToken/text()"></assign>
        <assign to="URI">/</assign>
        <assign to="RawResponse">true</assign>
        <assign to="ResponseTimeout">1200</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientGetServiceResults" from="*" append="true"></assign>
      </input>
    </operation>
    <operation name="HTTP Client End Session Service">
```

```
                    <participant name="HTTPClientEndSession"/>
                    <output message="HTTPClientEndSessionServiceTypeInputMessage">
                      <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
                            SessionToken/text()"></assign>
                      <assign to="." from="*"></assign>
                    </output>
                    <input message="inmsg">
                      <assign to="HTTPClientEndSessionServiceResults" from="*" append="true">
                          </assign>
                    </input>
                  </operation>
                  <onFault>
                    <sequence name="End Session">
                      <operation name="HTTP Client End Session Service">
                        <participant name="HTTPClientEndSession"/>
                        <output message="HTTPClientEndSessionServiceTypeInputMessage">
                          <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
                                SessionToken/text()"></assign>
                          <assign to="." from="*"></assign>
                        </output>
                        <input message="inmsg">
                          <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
                        </input>
                      </operation>
                    </sequence>
                  </onFault>
                </sequence>
            </process>
```

The following business process is an example of using the URI parameter in the
HTTP Client GET service. The URI is:

/s/ref=nb_ss_b/102-0129027-9554536?url=search-alias%3Dstripbooks&field-
keywords=Computers&Go.x=6&G0.y=4

This URI has the following parameters:

| Parameter | Description |
|---|---|
| url | search-alias%3Dstripbooks |
| field-keywords | Computers |
| Go.x | 6 |
| G0.y | 4 |

```
            <process name = "test_http_get">
              <sequence>
                <operation name="HTTP Client Begin Session Service">
                  <participant name="HTTPClientBeginSession"/>
                  <output message="HTTPClientBeginSessionServiceTypeInputMessage">
                    <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
                    <assign to="RemoteHost">www.amazon.com</assign>
                    <assign to="RemotePort">80</assign>
                    <assign to="." from="*"></assign>
                  </output>
                  <input message="inmsg">
                    <assign to="." from="*"></assign>
                  </input>
                </operation>
                <operation name="HTTP Client GET Service">
                  <participant name="HTTPClientGet"/>
                  <output message="HTTPClientGetServiceTypeInputMessage">
                     <assign to="SessionToken" from="/ProcessData/SessionToken/text()"></assign>
                    <assign to="URI">/s/ref=nb_ss_b/
                          102-0129027-9554536?url=search-alias%3Dstripbooks&amp;field-keywords=
```

```
            Computers&amp;Go.x=6&amp;Go.y=4</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken" from="/ProcessData/SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

The resulting document in process data is the link to the Amazon page with the search requested.

## Activity Types for the HTTP Client GET Service

The HTTP Client GET service reports the following activities to the Services Controller for service/GET service monitoring:

- GET – Retrieves whatever information is identified by the URI in the request line.

# Chapter 104. HTTP Client Method Service

The following table provides an overview of the HTTP Client Method service:

| System name | HTTP Client Method Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > HTTP Client |
| Description | The HTTP Client Method service sends HTTP requests to a trading partner's HTTP server via the perimeter server. The HTTP Client Method Service supports POST, GET, HEAD, or other valid HTTP request types that follow the normal HTTP request/response model. The actual HTTP methods available depends on the support provided by the remote/origin server. This service works in conjunction with the HTTP Client Begin service and the HTTP Client End service, and through an instance of the HTTP Client adapter. |
| Business usage | A business user would use the HTTP Client Method service to either retrieve or send data from Sterling Integrator to a trading partner when the HTTP protocol is required as the transport mechanism. |
| Usage example | A Sterling Integrator business process is executed that generates a document that must be sent to a trading partner using HTTP. Sterling Integrator initiates a session with the trading partner using the HTTP Client adapter and uses the HTTP Client Method Service, to place the document on the trading partner's HTTP server. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | Related services:<br>• HTTP Client adapter<br>• HTTP Client Begin Session service<br>• HTTP Client End Session service<br>• HTTP Client GET service<br>• HTTP Client POST service |
| Application requirements | An HTTP server at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• 100s – Informational parameters and more status parameters will follow.<br>• 200s – Request was OK.<br>• 300s – Request failed and more status codes will follow.<br>• 400s – Client Request Failed. This is a client error.<br>• 500s – Server failed to handle a valid request. This is a server error. |

| | |
|---|---|
| Restrictions | None |
| Persistence level | System Default |
| Testing considerations | To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the *HTTP Client adapter*. Debug information for this service can be found in the HTTP Client adapter and services log files. |

## Implementing the HTTP Client Method Service

To implement the HTTP Client Method service, complete the following tasks:

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

## Configuring the HTTP Client Method Service

To configure the HTTP Client Method service, you must specify field settings in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| DocumentId | A document ID to use with this instance of the HTTP Client Method service. Optional. |
| MethodType | Specifies the HTTP request type. Required. The HTTP Client Method Service supports POST, GET, HEAD, or other valid HTTP request types that follow the normal HTTP request/response model. The actual HTTP methods available depends on the support provided by the remote/origin server. |
| RawRequest | Whether to include the HTTP header metadata in the primary document. Optional. Valid values: <br>• true – HTTP header metadata is included in the primary document. If set to true, the request document. *must* contain headers inside its message body. <br>• false – HTTP header metadata is not included in the primary document. The headers are not available to the business process. <br><br>Default is false. <br>**Note:** This parameter does not apply if MethodType is set to GET. |

| Field | Description |
|-------|-------------|
| RawResponse | Whether to include the HTTP Server response headers in the response document. Optional. Valid values:<br>• true – Both the HTTP headers and the entity body are copied to the body of the business process document.<br>• false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process.<br>Default is false. |
| ResponseTimeout | Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.<br>**Note:** The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration. |
| SessionToken | Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required. |
| ShowResponseCode | Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:<br>• true – Metadata will be included.<br>• false – Metadata will not be included. Default is false. |
| URI | Uniform Resource Indicator (URI) representing the HTTP server. Required. |
| Cookie | Identifies the cookie to use in cases where the previous POST or GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Method Service will use the Cookie parameter and create the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client Method service:

| Field | Description |
|-------|-------------|
| DocumentId | A document ID to use with this instance of the HTTP Client Method service. Optional. |
| MethodType | Specifies the HTTP request type. Required. The HTTP Client Method Service supports POST, GET, HEAD, or other valid HTTP request types that follow the normal HTTP request/response model. The actual HTTP methods available depends on the support provided by the remote/origin server. Valid values include:<br>• GET – Retrieves whatever information is identified by the URI in the request line.<br>• POST – Requests that the server accept the entity enclosed in the request as a new subordinate of the resource identified by the URI in the request line.<br>• HEAD – Retrieves the header of whatever information is identified by the URI in the request line. |

| Field | Description |
|-------|-------------|
| RawRequest | Whether to include the HTTP header metadata in the primary document. Optional. Valid values: <br><br>• true – HTTP header metadata is included in the primary document. If set to true, the request document. *must* contain headers inside its message body. <br><br>• false – HTTP header metadata is not included in the primary document. The headers are not available to the business process. <br><br>Default is false. <br>**Note:** This parameter does not apply if MethodType is set to GET. |
| RawResponse | Whether to include the HTTP Server response headers in the response document. Optional. Valid values: <br><br>• true – Both the HTTP headers and the entity body are copied to the body of the business process document. <br><br>• false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process. <br><br>Default is false. |
| ResponseTimeout | Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid. <br>**Note:** The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration. |
| SessionToken | Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required. |
| ShowResponseCode | Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values: <br><br>• true – Metadata will be included. <br><br>• false – Metadata will not be included. Default is false. |
| URI | Uniform Resource Indicator (URI) representing the HTTP server. Required. |
| Cookie | Identifies the cookie to use in cases where the previous POST or GET needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Method Service will use the Cookie parameter and create the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional. |

## Business Process Examples

The following example business process illustrates using commands supported by the HTTP Client Method service:

```
 <process name="HTTPClientMethodServiceExample">
  <sequence>
⌂[[Insert Begin session operation here]]
    <operation name="HTTP Client Method Service">
      <participant name="HTTPClientMethodService"/>
      <output message="HTTPClientMethodServiceTypeInputMessage">
```

```
          <assign to="DocumentId">DocumentID</assign>
          <assign to="MethodType">HEAD</assign>
          <assign to="RawRequest">true</assign>
          <assign to="ResponseTimeout">60</assign>
          <assign to="SessionToken">SessionToken</assign>
          <assign to="ShowResponseCode">true</assign>
          <assign to="URI">URI</assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
⌂[[Insert End session operation here]]
    </sequence>
</process>
```

The following business process provides an example of using the Cookie
parameter:

```
<process name="HTTP_To_advancepcsrx">
  <!-- Loop Invariant. -->
  <rule name="haveMoreCookie1">
 <condition> string(counterCookie) &lt;= count(HTTPClientPostServiceResults/
        ServerResponse/Headers/Set-Cookie/node()) </condition>
  </rule>
  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="CACertificateId">sgmillenia:13582d:10682043f1d:-73bd</assign>
         <assign to="CipherStrength">all</assign>
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="SSL">Must</assign>
        <assign to="RemoteHost">webtransport.advancepcsrx.com</assign>
        <assign to="RemotePort">443</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>
    <!-- Use FSA to pick up the input file -->
    <operation name="Import Document Request">
      <participant name="TEST_FILE_SYSTEM_ADAPTER"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder" from="'/ais_local/share/sli'"></assign>
        <assign to="filter" from="'AdvancePCS_URI.txt'"></assign>
        <assign to="useSubFolders">false</assign>
        <assign to="bootstrap">false</assign>
        <assign to="deleteAfterCollect">false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="FileSystemOutputMessage">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <!-- Set document content type/subtype -->
    <operation name="SetContenType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="." from="*"></assign>
        <assign to="DocumentContentType">application</assign>
        <assign to="DocumentContentSubType">x-www-form-urlencoded</assign>
      </output>
      <input message="xin">
```

```
            <assign to="docInfo" from="*"></assign>
          </input>
        </operation>
        <!-- POST to URI /template/login to login the webpage -->
        <operation name="Http Client Post Service">
          <participant name="HTTPClientPost"/>
          <output message="HTTPClientPostServiceTypeInputMessage">
            <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
                SessionToken/text()"></assign>
            <assign to="URI">/template/login</assign>
            <assign to="RawResponse">true</assign>
            <assign to="ResponseTimeout">120</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="HTTPClientPostServiceResults" from="*" append="true"></assign>
          </input>
        </operation>
        <!-- concat the cookie from the POST Response -->
        <assign to="counterCookie">1</assign>
        <assign to="Cookie" from="''"/>
        <choice>
<select>
<case ref="haveMoreCookie1" activity="AppendCookie1"/>
</select>
<sequence name="AppendCookie1">
<assign to="Cookie" from="concat(string(Cookie), substring-before
        (//HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie
        [number(//counterCookie)],';'), '; ')"/>
<assign to="counterCookie" from="number(counterCookie) + 1"/>
<choice>
<select>
<case ref="haveMoreCookie1" activity="Repeat1"/>
</select>
<repeat name="Repeat1" ref="AppendCookie1"/>
</choice>
</sequence>
        </choice>
        <!-- release Primary Document before GET -->
        <operation>
<participant name="ReleaseService"/>
<output message="releaseRequest">
<assign to="TARGET" from="'PrimaryDocument'"/>
</output>
<input message="releaseResponse"/>
        </operation>
        <!-- GET URI / to reach the final page -->
        <operation name="Http Client Get Service">
          <participant name="HTTPClientGet"/>
          <output message="HTTPClientGetServiceTypeInputMessage">
            <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
                SessionToken/text()"></assign>
            <assign to="URI">/</assign>
            <assign to="RawResponse">true</assign>
            <assign to="ResponseTimeout">1200</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="HTTPClientGetServiceResults" from="*" append="true"></assign>
          </input>
        </operation>
        <operation name="HTTP Client End Session Service">
          <participant name="HTTPClientEndSession"/>
          <output message="HTTPClientEndSessionServiceTypeInputMessage">
            <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
                SessionToken/text()"></assign>
            <assign to="." from="*"></assign>
```

```
            </output>
            <input message="inmsg">
              <assign to="HTTPClientEndSessionServiceResults" from="*" append="true">
                </assign>
            </input>
          </operation>
          <onFault>
            <sequence name="End Session">
              <operation name="HTTP Client End Session Service">
                <participant name="HTTPClientEndSession"/>
                <output message="HTTPClientEndSessionServiceTypeInputMessage">
                  <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
                        SessionToken/text()"></assign>
                  <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
                  <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
                </input>
              </operation>
            </sequence>
          </onFault>
        </sequence>
</process>
```

The following business process provides an example of using the
GetDocumentInfoService to set the content-type/subtype of the request document.
It is important to put the output parameters of GetDocumentInfoService into a
parent node, such as "docInfo".

The GetDocumentInfoService returns a node called DocumentId to the process
data. This is the ID of the original document object without the content-type and
sub-type set. If this node is directly under the root of Process Data, the HTTP
Client Post service will attempt to retrieve the original document object using the
document ID instead of the Primary Document. If DocumentId node is returned
under "docinfo" node, the HTTP Client Post service does not use the document
from DocumentId. Instead the service retrieves the correct document that was set
with the correct content-type/subtype from the primary document.

```
<process name="HttpClient_SetContentType">
  <sequence>
    <!-- Set document content type/subtype -->
    <operation name="SetContenType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="." from="*"></assign>
        <assign to="DocumentContentType">text</assign>
        <assign to="DocumentContentSubType">xml</assign>
      </output>
      <input message="xin">
        <assign to="docInfo" from="*"></assign>
      </input>
    </operation>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">10.235.18.103</assign>
        <assign to="RemotePort">37133</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>
    <operation name="Http Client Method Service">
```

```
      <participant name="HTTPClientMethod"/>
      <output message="HTTPClientMethodServiceTypeInputMessage">
        <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
              SessionToken/text()"></assign>
        <assign to="MethodType">POST</assign>
        <assign to="URI">/hello</assign>
        <assign to="RawRequest">false</assign>
        <assign to="RawResponse">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientMethodServiceResults" from="*"
            append="true"></assign>
      </input>
    </operation>
    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
            SessionToken/text()"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientEndSessionServiceResults" from="*" append="true">
            </assign>
      </input>
    </operation>
    <onFault>
      <sequence name="End Session">
        <operation name="HTTP Client End Session Service">
          <participant name="HTTPClientEndSession"/>
          <output message="HTTPClientEndSessionServiceTypeInputMessage">
            <assign to="SessionToken" from="HTTPClientBeginSessionServiceResults/
                  SessionToken/text()"></assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
          </input>
        </operation>
      </sequence>
    </onFault>
  </sequence>
</process>
```

## Activity Types for the HTTP Client Method Service

The HTTP Client Method service reports the following activities to the Services
Controller for service/method service monitoring:

- GET – Retrieves whatever information is identified by the URI in the request
  line.
- POST – Requests that the server accept the entity enclosed in the request as a
  new subordinate of the resource identified by the URI in the request line.
- HEAD – Retrieves the header of whatever information is identified by the URI
  in the request line.

The actual HTTP methods available depends on the support provided by the
remote/origin server.

# Chapter 105. HTTP Client POST Service

The following table provides an overview of the HTTP Client POST service:

| System name | HTTP Client POST Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, B2B Protocols > HTTP Client |
| Description | The HTTP Client POST Service sends HTTP POST requests to a trading partner's HTTP server via the perimeter server. This service works in conjunction with the HTTP Client Begin service and the HTTP Client End service, and through an instance of the HTTP Client adapter. |
| Business usage | A business user would use the HTTP Client POST Service to place documents on a trading partner's HTTP server. |
| Usage example | A business process is executed that generates a document to be transferred to a trading partner using the HTTP protocol. The application initiates a session with the trading partner using the HTTP Client Adapter and uses the HTTP Client POST Service to place the data on the trading partner's HTTP server. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported platforms for Sterling Integrator |
| Related services | Related services: <br> • HTTP Client adapter <br> • HTTP Client Begin Session service <br> • HTTP Client End Session service <br> • HTTP Client GET service <br> • HTTP Client Method service |
| Application requirements | An HTTP server must exist at the external trading partner location. |
| Initiates business processes? | No |
| Invocation | This service is invoked from a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values: <br> • 100s – Informational parameters and more status parameters will follow. <br> • 200s – Request was OK. <br> • 300s – Request failed and more status codes will follow. <br> • 400s – Client Request Failed. This is a client error. <br> • 500s – Server failed to handle a valid request. This is a server error. |
| Restrictions | None |
| Persistence level | Default |

| | |
|---|---|
| Testing considerations | To test this service, run the HTTPClientDemoAllServices business process and verify that it completes successfully. For more information about the HTTPClientDemoAllServices business process, see the *HTTP Client adapter*. Debug information for this service can be found in the HTTP Client adapter and services log files. |

## Implementing the HTTP Client POST Service

To implement the HTTP Client POST service, complete the following tasks:

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

## Configuring the HTTP Client POST Service

To configure the HTTP Client POST service, you must specify field settings in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. |
| DocumentId | A document ID to use with this instance of the HTTP Client POST service. Optional. |
| RawRequest | Whether to include the HTTP header metadata in the primary document. Optional. Valid values:<br>• true – HTTP header metadata is included in the primary document. If set to true, the request document *must* contain headers inside its message body.<br>• false – HTTP header metadata is not included in the primary document. The headers are not available to the business process.<br>Default is false.<br>**Note:** This parameter does not apply if MethodType is set to GET. |
| RawResponse | Whether to include the HTTP Server response headers in the response document. Optional. Valid values:<br>• true – Both the HTTP headers and the entity body are copied to the body of the business process document.<br>• false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process.<br>Default is false. |
| ResponseTimeout | Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.<br>**Note:** The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration. |

| Field | Description |
|-------|-------------|
| SessionToken | Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required. |
| ShowResponseCode | Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:<br>• true – Metadata will be included.<br>• false – Metadata will not be included. Default is false. |
| URI | Uniform Resource Indicator (URI) representing the HTTP server. Required. |
| Cookie | Identifies the cookie to use in cases where the previous POST needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Post Service uses the Cookie parameter and creates the Cookie header in the next request. Valid values are cookies separated by semi-colon (cookie1;cookie2;cookie3). Optional. |

## Output from Business Process to Service

The following table contains the parameters passed from the business process to the HTTP Client POST service:

| Field | Description |
|-------|-------------|
| DocumentId | A document ID to use with this instance of the HTTP Client POST service. Optional. |
| RawRequest | Whether to include the HTTP header metadata in the primary document. Optional. Valid values:<br>• true – HTTP header metadata is included in the primary document. If set to true, the request document *must* contain headers inside its message body.<br>• false – HTTP header metadata is not included in the primary document. The headers are not available to the business process.<br>Default is false.<br>**Note:** This parameter does not apply if MethodType is set to GET. |
| RawResponse | Whether to include the HTTP Server response headers in the response document. Optional. Valid values:<br>• true – Both the HTTP headers and the entity body are copied to the body of the business process document.<br>• false – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process.<br>Default is false. |
| ResponseTimeout | Number of seconds the HTTP Client adapter will wait for a response. Optional. Any numeric value is valid.<br>**Note:** The value entered for this parameter overrides the Timeout setting in the HTTP Client adapter configuration. |

| Field | Description |
|---|---|
| SessionToken | Specifies the identifier for the session established between the HTTP Client Adapter and an HTTP server. Required. |
| ShowResponseCode | Whether to include the HTTP response status code as the first line in the primary document. Optional. Valid values:<br>• true – Metadata will be included.<br>• false – Metadata will not be included. Default is false. |
| URI | Uniform Resource Indicator (URI) representing the HTTP server. Required. |
| Cookie | Identifies the cookie to use in cases where the previous POST needs page redirection (returns 300 range response). The cookie value is passed from the response header Set-Cookie. The Post Service uses the Cookie parameter and creates the Cookie header in the next request. |

## Business Process Examples

The following example business process illustrates using commands supported by the HTTP Client POST service:

```
<process name="HTTPClientPOSTServiceExample">
 <sequence>
 [[Insert Begin session operation here]]
   <operation name="HTTP Client POST Service">
     <participant name="HTTPClientPOSTService"/>
     <output message="HTTPClientPostServiceTypeInputMessage">
       <assign to="DocumentId">DocumentID</assign>
       <assign to="RawRequest">true</assign>
       <assign to="RawResponse">true</assign>
       <assign to="ResponseTimeout">60</assign>
       <assign to="SessionToken">SessionToken</assign>
       <assign to="ShowResponseCode">true</assign>
       <assign to="URI">URI</assign>
       <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
       <assign to="." from="*"></assign>
     </input>
   </operation>
 [[Insert End session operation here]]
 </sequence>
</process>
```

The following business process provides an example of using the Cookie parameter:

```
<process name="HTTP_To_advancepcsrx">
 <!-- Loop Invariant. -->
 <rule name="haveMoreCookie1">
   <condition> string(counterCookie) &lt;=
     count(HTTPClientPostServiceResults/ServerResponse/Headers/
       Set-Cookie/node()) </condition>
 </rule>
 <sequence>
   <operation name="HTTP Client Begin Session Service">
     <participant name="HTTPClientBeginSession"/>
     <output message="HTTPClientBeginSessionServiceTypeInputMessage">
       <assign to="CACertificateId">sgmillenia:13582d:10682043f1d:-73bd
```

```
            </assign>
        <assign to="CipherStrength">all</assign>
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="SSL">Must</assign>
        <assign to="RemoteHost">webtransport.advancepcsrx.com</assign>
        <assign to="RemotePort">443</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </input>
    </operation>
    <!-- Use FSA to pick up the input file -->
    <operation name="Import Document Request">
      <participant name="TEST_FILE_SYSTEM_ADAPTER"/>
      <output message="FileSystemInputMessage">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder" from="'/ais_local/share/sli'"></assign>
        <assign to="filter" from="'AdvancePCS_URI.txt'"></assign>
        <assign to="useSubFolders">false</assign>
        <assign to="bootstrap">false</assign>
        <assign to="deleteAfterCollect">false</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="FileSystemOutputMessage">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <!-- Set document content type/subtype -->
    <operation name="SetContenType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="." from="*"></assign>
        <assign to="DocumentContentType">application</assign>
        <assign to="DocumentContentSubType">x-www-form-urlencoded</assign>
      </output>
      <input message="xin">
        <assign to="docInfo" from="*"></assign>
      </input>
    </operation>
    <!-- POST to URI /template/login to login the webpage -->
    <operation name="Http Client Post Service">
      <participant name="HTTPClientPost"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="SessionToken"
         from="HTTPClientBeginSessionServiceResults/SessionToken/text()">
            </assign>
        <assign to="URI">/template/login</assign>
        <assign to="RawResponse">true</assign>
        <assign to="ResponseTimeout">120</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientPostServiceResults"
             from="*" append="true"></assign>
      </input>
    </operation>
    <!-- concat the cookie from the POST Response -->
    <assign to="counterCookie">1</assign>
    <assign to="Cookie" from="''"/>
    <choice>
<select>
<case ref="haveMoreCookie1" activity="AppendCookie1"/>
</select>
<sequence name="AppendCookie1">
<assign to="Cookie" from="concat(string(Cookie), substring-before
  (//HTTPClientPostServiceResults/ServerResponse/Headers/Set-Cookie[number
```

```
                    (//counterCookie)],';'),'; ')"/>
        <assign to="counterCookie" from="number(counterCookie) + 1"/>
        <choice>
        <select>
        <case ref="haveMoreCookie1" activity="Repeat1"/>
        </select>
        <repeat name="Repeat1" ref="AppendCookie1"/>
        </choice>
        </sequence>
            </choice>
        <!-- release Primary Document before GET -->
            <operation>
        <participant name="ReleaseService"/>
        <output message="releaseRequest">
        <assign to="TARGET" from="'PrimaryDocument'"/>
        </output>
        <input message="releaseResponse"/>
            </operation>
            <!-- GET URI / to reach the final page -->
            <operation name="Http Client Get Service">
              <participant name="HTTPClientGet"/>
              <output message="HTTPClientGetServiceTypeInputMessage">
                <assign to="SessionToken"
                 from="HTTPClientBeginSessionServiceResults/SessionToken/text()">
                    </assign>
                <assign to="URI">/</assign>
                <assign to="RawResponse">true</assign>
                <assign to="ResponseTimeout">1200</assign>
                <assign to="." from="*"></assign>
              </output>
              <input message="inmsg">
                <assign to="HTTPClientGetServiceResults" from="*" append="true">
                    </assign>
              </input>
            </operation>
            <operation name="HTTP Client End Session Service">
              <participant name="HTTPClientEndSession"/>
              <output message="HTTPClientEndSessionServiceTypeInputMessage">
                <assign to="SessionToken"
                 from="HTTPClientBeginSessionServiceResults/SessionToken/text()">
                    </assign>
                <assign to="." from="*"></assign>
              </output>
              <input message="inmsg">
                <assign to="HTTPClientEndSessionServiceResults" from="*"
                 append="true"></assign>
              </input>
            </operation>
                    <onFault>
            <sequence name="End Session">
              <operation name="HTTP Client End Session Service">
                <participant name="HTTPClientEndSession"/>
                <output message="HTTPClientEndSessionServiceTypeInputMessage">
                  <assign to="SessionToken"
                 from="HTTPClientBeginSessionServiceResults/SessionToken/text()">
                    </assign>
                  <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
                  <assign to="HTTPClientEndSessionServiceResults" from="*"></assign>
                </input>
              </operation>
            </sequence>
            </onFault>
          </sequence>
        </process>
```

The following business process provides an example of using the GetDocumentInfoService to set the content-type/subtype of the request document. It is important to put the output parameters of GetDocumentInfoService into a parent node, such as "docInfo".

The GetDocumentInfoService returns a node called DocumentId to the process data. This is the ID of the original document object without the content-type and sub-type set. If this node is directly under the root of Process Data, the HTTP Client Post service will attempt to retrieve the original document object using the document ID instead of the Primary Document. If DocumentId node is returned under "docinfo" node, the HTTP Client Post service does not use the document from DocumentId. Instead the service retrieves the correct document that was set with the correct content-type/subtype from the primary document.

```
<process name="HttpClient_SetContentType">
  <sequence>
    <!-- Set document content type/subtype -->
    <operation name="SetContenType">
      <participant name="GetDocumentInfoService"/>
      <output message="xout">
        <assign to="." from="*"></assign>
        <assign to="DocumentContentType">text</assign>
        <assign to="DocumentContentSubType">xml</assign>
      </output>
      <input message="xin">
        <assign to="docInfo" from="*"></assign>
      </input>
    </operation>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">10.235.18.103</assign>
        <assign to="RemotePort">37133</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientBeginSessionServiceResults" from="*"></assign>
      </ input>
    </operation>
    <operation name="Http Client Post Service">
      <participant name="HTTPClientPost"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="SessionToken"
         from="HTTPClientBeginSessionServiceResults/SessionToken/text()">
            </assign>
        <assign to="URI">/hello</assign>
        <assign to="RawRequest">false</assign>
        <assign to="RawResponse">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="HTTPClientPostServiceResults" from="*" append="true">
            </assign>
      </input>
    </operation>
    <operation name="HTTP Client End Session Service">
      <participant name="HTTPClientEndSession"/>
      <output message="HTTPClientEndSessionServiceTypeInputMessage">
        <assign to="SessionToken"
          from="HTTPClientBeginSessionServiceResults/SessionToken/text()">
            </assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
```

```
                              <assign to="HTTPClientEndSessionServiceResults" from="*"
                               append="true"></assign>
                          </input>
                        </operation>
                       <onFault>
                         <sequence name="End Session">
                           <operation name="HTTP Client End Session Service">
                             <participant name="HTTPClientEndSession"/>
                             <output message="HTTPClientEndSessionServiceTypeInputMessage">
                               <assign to="SessionToken"
                              from="HTTPClientBeginSessionServiceResults/SessionToken/text()">
                                     </assign>
                               <assign to="." from="*"></assign>
                             </output>
                             <input message="inmsg">
                               <assign to="HTTPClientEndSessionServiceResults" from="*">
                                     </assign>
                             </input>
                           </operation>
                         </sequence>
                       </onFault>
                      </sequence>
                    </process>
```

The following business process provides an example of specifying parameters in
URI of the HTTP Client POST service. The URI is:

```
/getraf/portal_getraf/processGentranData?
              UID=f87db70048484b0fe6348eaebbf62281&status=0&errorMsg=Erro
```

The URI in this example has three parameters:

| Parameter | Value |
|-----------|-------|
| ei | utf-8 |
| fr | slv8-msgr |
| p | http%20POST%20examples |

The business process is the following:

```
<process name = "test_http_post">
  <sequence>
    <operation name="HTTP Client Begin Session Service">
      <participant name="HTTPClientBeginSession"/>
      <output message="HTTPClientBeginSessionServiceTypeInputMessage">
        <assign to="HTTPClientAdapter">HTTPClientAdapter</assign>
        <assign to="RemoteHost">search.yahoo.com</assign>
        <assign to="RemotePort">80</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="HTTP Client POST Service">
      <participant name="HTTPClientPost"/>
      <output message="HTTPClientPostServiceTypeInputMessage">
        <assign to="RawRequest">true</assign>
        <assign to="SessionToken" from="/ProcessData/SessionToken/text()">
              </assign>
        <assign to="ShowResponseCode">true</assign>
        <assign to="URI">/search?ei=utf-8&amp;fr=slv8-msgr&amp;p=
              http%20POST%20examples</assign>
        <assign to="." from="*"></assign>
      </output>
```

```
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
    </operation>
    <operation name="HTTP Client End Session Service">
        <participant name="HTTPClientEndSession"/>
        <output message="HTTPClientEndSessionServiceTypeInputMessage">
          <assign to="SessionToken" from="/ProcessData/SessionToken/text()">
              </assign>
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
    </operation>
</sequence>
</process>
```

## Activity Types for the HTTP Client POST Service

The HTTP Client POST service reports the following activities to the Services
Controller for service/POST service monitoring:

- POST – Requests that the server accept the entity enclosed in the request as a
  new subordinate of the resource identified by the URI in the request line.

# Chapter 106. HTTP Respond Service

The following table provides an overview of the HTTP Respond service:

| System name | HTTP Respond Service |
|---|---|
| Graphical Process Modeler (GPM) category | None |
| Description | This adapter sends responses to trading partners who have submitted an HTTP request. The request could have been submitted through the B2B HTTP Server adapter or through the new, Perimeter server-based HTTP Server adapter.<br>**Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. For more information on the retirement process, see *Retiring and Removed Services and Adapters*. |
| Business usage | A business user would use this adapter to return a response to a trading partner over the HTTP transport protocol, when the trading partner initiates the HTTP connection. |
| Usage example | A trading partner submits a document to a URL in Sterling Integrator. The URL is configured to initiate a specific business process. The business process completes activities and returns a response to the trading partner to confirm that the request was processed. This response is returned using this service. |
| Preconfigured? | Yes |
| Requires third party files? | When used with the HTTP Server adapter, Jetty's two jar files must be available; Jetty is an open source, embeddable webserver and servlet engine (http://jetty.mortbay.org). |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | HTTP Server adapter |
| Application requirements | Unless the Perimeter server local-mode is used, the external Perimeter server must be running. |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Business process context considerations | No |
| Returned status values | Returned status values:<br>• Success – Normal execution with the return parameters specified below.<br>• Error – For any error cases, namely missing parameter and adapter not found. |
| Restrictions | None |
| Persistence level | None |

| | |
|---|---|
| Testing considerations | Debug information related to this adapter can be found in http.log. The level of logging can be controlled through the Sterling Integrator UI or the log.properties file. |

## How the HTTP Respond Service Works

The following example illustrates how the HTTP Respond service works:

1. A trading partner sends data to your company using HTTP.
2. Your HTTP Server adapter receives the data, initiates a business process, and writes the transport-instance-id and transport-session-id for the HTTP connection into process data.
3. The business process runs and after specified tasks are completed, invokes the HTTP Respond service.
4. The business process passes the transport-instance-id and transport-session-id from the original HTTP connection to the service.
5. The service uses these IDs to find the original HTTP connection, which is still open, and uses it to send a reply to your trading partner.

## Implementing the HTTP Respond Service

A configuration of the HTTP Respond service (HttpRespond) is provided with Sterling Integrator. Because it has no configuration parameters, it should not be necessary to create any new configurations of the HTTP Respond service. The configuration provided should suffice for all uses of this service, including using the service in a business process.

## Output from Business Process to Service

The following table describes the output from the business process to the HTTP Respond service:

| Parameter | Description |
|---|---|
| transport-instance-id | Identifies the specific instance of the adapter on which the request was received. This might be a B2B HTTP Server adapter or a Perimeter server-based HTTP Server adapter. Valid value is a non-empty string generated by an HTTP server adapter. Required.<br>**Note:** Because of our continuing efforts to improve services and adapters to align with new technology and capabilities, the B2B HTTP Server adapter has entered the retirement process in Sterling Integrator and will be replaced with the HTTP Server adapter. For more information on the retirement process, see *Retiring and Removed Services and Adapters*. |
| transport-session-id | Transport Session ID. Identifies the specific inbound connection on the HTTP Server adapter on which the request was received and to which the response must be returned. In either case, the adapter is identified by the transport instance ID. Valid value is a non-empty string that should not be fabricated. Each value is created by an HTTP Server adapter instance for an inbound HTTP session. Required. |

| Parameter | Description |
|---|---|
| doc-has-headers | Raw mode selection for the response. Valid values are True and False. If True, when the service runs, the primary document is expected to have the HTTP headers for the response; that is, the primary document has a MIME, RFC822 or HTTP structure with headers and body. If False, the primary document does not contain any headers. Required. |
| status-code | The HTTP status code for the response message to the HTTP client. Each range has a specific meaning. For example, codes between 200 and 299 are success codes and those between 400 and 599 specify different kinds of failures. Valid values are between 100 and 599. Default is 200. Optional. |

## Business Process Example

The following example business process returns an HTTP response with the contents of the primary document being the HTTP body (non-raw mode):

```
<process name="NonRawHttpRespond">
  <sequence>
    <operation>
      <participant name="HttpRespond"/>
      <output message="noopout">
        <assign to="doc-has-headers">false</assign>
        <assign to="." from="*"/>
      </output>
      <input message="noopin">
        <assign to="." from="*"/>
      </input>
    </operation>
  </sequence>
</process>
```

# Chapter 107. HTTP Reverse Proxy Adapter

The HTTP Reverse Proxy adapter provides a high level of data protection between external connections using HTTP and your company's Sterling Integrator server. Use this adapter to receive documents from a trading partner who uses the HTTP protocol.

The HTTP Reverse Proxy adapter can only be used with the Sterling Secure Proxy 2.0 (or later) product. See the Sterling Secure Proxy documentation for information.

# Chapter 108. HTTP Server Adapter

The following table provides an overview of the HTTP Server adapter.

| System name | HTTP Server Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | None. Cannot be used as part of a business process. |
| Description | Processes HTTP requests from trading partners using a Perimeter server. The HTTP Server adapter includes a bundled WebDAV server that is licensed independently. For information about using WebDAV in your application, see the *WebDAV Server* documentation. The HTTP Server adapter replaces the B2B HTTP Server adapter, which is being retired. The HTTP Server adapter and its related services provide all of the functionality of that adapter, plus these benefits: <br><br>• Uses Perimeter services <br>• Uses the same Jetty HTTP server engine as the application ASI console <br>• Able to run both WARs and BPML applications <br>• Runs application code inside the JVM for access to all application resources |
| Business usage | Use this adapter to send documents to and receive documents from a trading partner using HTTP. |
| Usage example | A trading partner submits a document to a URL in the application. The URL is configured to initiate a specific business process. The business process completes activities and returns a success response to the trading partner to confirm that the request was processed. If the URL is configured for a Web application, the identified Web application is started and delegated to. At that point the flow is determined by the Web application. |
| Preconfigured? | Eight instances of this adapter are installed with a standalone instance of the application. By default, they use a local-mode Perimeter server. They are: ebXML HTTP Server adapter, HTTP Server adapter, Map Test HTTP Server adapter, MBI HTTP Server adapter, RN HTTP Server adapter, SOA HTTP Server adapter, SOA SSL HTTP server adapter, and SWIFTNet HTTP Server adapter. A two-node clustered environment consisting of nodes on different machines includes an instance of HttpServer adapter node2. |
| Requires third party files? | Must obtain two.jar files from Jetty. Jetty is an open source, embeddable Web server and servlet engine (http://jetty.mortbay.org). Two .jar files for Jasper (an open source JSP compilation library) and the. jar file for Ant (an open source build and compilation library) must be available for the handling of Web applications that contain JSPs. |
| Platform availability | All supported application platforms. |
| Related services | This adapter must be used in conjunction with the HTTP Respond service, which is the only way to return an HTTP response to a request waiting at a particular adapter instance. |

| Application requirements | When this adapter is configured with a non-local-mode Perimeter server, the Perimeter server must be installed and running. This Perimeter server is typically installed in a DMZ environment, separated from the application by a firewall. |
|---|---|
| Initiates business processes? | This adapter can find the name of a business process that is configured for a particular URL, initiate that business process and wait for the response. |
| Invocation | Is not invoked by a business process. To return a response, use the HTTP Respond service. |
| Business process context considerations | When a business process is initiated as a result of an HTTP request, the initial context process data contains the transport-instance-id and transport-session-id, information necessary for the HTTP Respond service to return the HTTP response. Process data also contains any query parameters in the URL. |
| Returned status values | None |
| Restrictions | WAR file deployment functionality is not available for WebSphere. |
| Persistence level | None |
| Testing considerations | Debug information related to this adapter can be found in http.log. |

## How the HTTP Server Adapter Works

The HTTP Server adapter receives data from a trading partner using HTTP. Using the HTTP Server adapter, you can initiate a business process or a Web application. If you are initiating a business process and want to send a reply or status to your trading partner about this data, you must have the business process invoke the HTTP Respond service, which sends the response to your trading partner.

**Note:** This adapter differs from the previous HTTP adapter (B2B HTTP Server adapter), which is in the retirement process, in how it works with secure installations that use a DMZ. The older B2B HTTP Server adapter requires an HTTP servlet to be installed in your company DMZ. Instead of using a servlet, this adapter communicates with a Perimeter server that has been installed in the DMZ. For more information on the retirement process, see *Retiring and Removed Services and Adapters*.

## Example

Your trading partner sends EDI purchase orders to your company, using HTTP and sending the data to a URI that you have specified.

You set up a configuration of the HTTP Server adapter that is associated with that URI. As part of creating this configuration, you specify whether the URI will be associated with a business process or a Web application, then select the specific business process or web application.

Your trading partner sends data to your URI, using an HTTP request. When the request is received and passed to the HTTP Server adapter, it invokes the business process or Web application that you specified when creating the adapter

configuration. The adapter collects the transport-instance-id and transport-session-id from the initial request, and places the information into process data.

If the adapter initiates a business process, the HTTP request connection is put into a wait state while the business process completes. Once complete, the HTTP Respond service is called and uses the transport-instance-id and transport-session-id that were stored in process data to send a reply on the same connection that the request came in on.

If the adapter initiates a Web application, the Web application determines what to return and when.

## Implementing the HTTP Server Adapter

To implement the HTTP Server adapter, complete the following tasks:

1. Create an HTTP Server adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the HTTP Server adapter. For information, see *Configuring the HTTP Server Adapter*.

   **Note:** If configuring the HTTP Server adapter for WebDAV, see the specific configuration instructions in the WebDAV Server documentation.

## Configuring the HTTP Server Adapter

To configure the HTTP Server adapter, you must specify field settings in the application, as described in the following table.

| Field | Description |
|---|---|
| Name | Unique and meaningful name for the adapter configuration. Required. |
| Description | Meaningful description for the adapter configuration, for reference purposes. Required. |
| Select a Group | Leave this set to:<br>• None – You do not want to include this configuration in a group at this time.<br><br>**Note:** Do not use the HTTP Server adapter in groups. |
| HTTP Listen Port | The port number on which the Perimeter server process listens for connections from external trading partner HTTP clients. If a local-mode Perimeter server is chosen, this listen port is bound on the local computer. Valid values are 1 through 65536. On many operating systems, only the root user can bind on ports 1 through 1024. Required. |
| Perimeter Server Name | List of available Perimeter servers, including local-mode Perimeter servers. Required. Default is local-mode Perimeter server. |

| Field | Description |
|---|---|
| Total Business Process queue depth threshold: | Indicates the maximum number of queued business processes allowed for this adapter. At this point, system load is too high to accept a new business process invoke request. If a value other than 0 is specified, the adapter will limit the number of business process requests put on the queue. If the SUM of business processes on all queues is less than the queue threshold value, processing occurs normally. For example, a queue threshold of 500 will stop a request if queue 4 has 300 business processes, queue 6 has 200, and queue 7 has 3. If the threshold is exceeded, the adapter returns a Service Unavailable message, which triggers senders to retry later. Valid value is any integer. 0 indicates no threshold (feature turned off). |
| Document Storage | Where to store the body of the request document. Valid values are:<br>• System Default<br>• Database<br>• File System<br><br>Default is System Default. Required.<br>**Note:** For more information about document storage types, see *Managing Services and Adapters*. |
| User Authentication Required | Whether to enable HTTP basic authentication. Valid values are:<br>• Yes – A connection must pass HTTP basic authentication to be serviced.<br>• No – HTTP basic authentication is not to be used.<br><br>Default is Yes. Required. |
| Use SSL | Whether SSL Server authentication must be enabled or not. Valid values are:<br>• Must – SSL is enabled<br>• None – SSL is disabled<br><br>Default is None. Required. |
| System Certificate | Select a system certificate from the list. This is the private key that the SSL server will use. Required if Use SSL is Must. |
| Cipher Strength | Specifies the strength of the algorithms (cipher suites) used to encrypt data. Valid values are:<br>• STRONG – Required if Use SSL is Must<br>• ALL – All cipher strengths are supported<br>• WEAK – Often required for international trade, because government regulations prohibit STRONG encryption from being exported<br><br>Default is STRONG. Required if SSL is checked. |
| CA Certificate | Move one or more CA Certificates to the use column. These are the digital security certificates that the SSL server will use to authenticate the client. Optional. |

| Field | Description |
|---|---|
| URI | Uniform Resource Indicator (URI) representing incoming requests. Add one or more URIs to represent incoming requests and the business process or web application (represented as a WAR file) associated with each. Required. |
| Launch BP or WAR | Whether the URI launches a business process or WAR file. Default is Business Process. Required. |
| Enter WAR File Path | Specifies WAR file to be launched by URI. Valid value is any accessible path. Required if WAR File is selected for Launch BP or WAR File field. |
| Business Process | Specifies business process to be launched by URI. Select from the list of available business processes. Required if BP is selected for Launch BP or WAR File field. |
| Send Raw Messages | Whether the raw message is presented to the business process. The term raw denotes that the primary document associated with the business process contains HTTP headers. Valid values are:<br><br>• Yes – Both the HTTP headers and the entity body are copied to the body of the business process document before the business process is started. This setting is required for EDIINT AS2, RosettaNet, and ebXML.<br><br>• No – Just the HTTP entity body is copied to the body buffer of the business process document. The headers are not available to the business process.<br><br>Default is No. Required if BP is selected for Launch BP or WAR File field. |
| Run BP in sync mode | Whether to invoke Web services in synchronous mode. Valid values are:<br><br>• Yes – HTTP Server adapter bootstraps the BP in synchronous mode. HTTP Server adapter executes the BP in the same thread.<br><br>• No – HTTP Server adapter bootstraps the BP asynchronous mode.<br><br>Default is No. Required if BP is to be run in synchronous mode. |

## Output from Adapter to Business Process

The following table describes the output from the HTTP Server adapter to the business process.

| Field Name | Description |
|---|---|
| http-request-uri | Target URI as specified by the Trading Partner. Required for applications that need it. For example, SOAP. |
| transport-instance-id | Identifies the specific instance of the HTTP Server adapter on which the request was received. Valid value is a non-empty string that should not be fabricated. Each value is created for an HTTP Server adapter instance. Required. |

| Field Name | Description |
|---|---|
| transport-session-id | Transport Session ID. Identifies the specific inbound connection on the HTTP Server adapter (identified by the transport instance id) on which the request was received and to which the response must be returned. Valid value is a non-empty string that should not be fabricated. Each value is created by an HTTP Server adapter instance for an inbound HTTP session. Required. |
| b2b-protocol | Identifies the protocol type, with a value of **http.** |
| SyncModeBP | Indicates whether the BP has been bootstrapped in synchronous mode or not. Valid values are true and false. |

## Initial Process Data XML Document Example

The following example shows how the initial process data XML document looks:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
  <PrimaryDocument SCIObjectID="server1:b1aebf:fa40ae79ca:-7209"/>
  <b2b-protocol>http</b2b-protocol>  △
     <transport-instance-id>TestHTTPServerAdapter-insecure_HttpAdapter_node1
     </transport-instance-id>
  <transport-session-id>Thu Jan 22 22:04:16 EST 2004:5</transport-session-id>
△<http-request-uri>/reflect</http-request-uri>
</ProcessData>
```

## Activity Types for this Service

This adapter reports the following activities to the Services Controller for Service/Adapter Monitoring:

- Get – Retrieves whatever information is identified by the http-request-uri
- Post – Requests that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the URI in the Request-Line .

## Enhancing HTTP Server Adapter Performance

To improve performance, the HTTP Server adapter enables you to specify a range of threads for handling events. The range, which is specified in the http.properties file, includes a Min Thread value and Max Thread value. If the Max Thread value is reached, any additional connection requests fail. The http.properties file is located in the properties folder under your application installation directory.

# Chapter 109. IBM Information Exchange FTP (IBM IE FTP) Adapter

The following table provides an overview of the IBM Information Exchange FTP (IBM IE FTP) adapter, which is used to communicate with the IBM Information Exchange network:

| System name | IBM IE FTP adapter |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | The IBM IE FTP adapter is used to send and collect documents from the IBM EDI services mailbox using a TCP/IP FTP gateway. |
| Business usage | Used to exchange documents with a trading partner's IBM EDI services mailbox. |
| Usage example | An internal service invokes a business process that sends a trading document to a trading partner mailbox in the IBM VAN. |
| Preconfigured? | No |
| Requires third party files? | None |
| Platform availability | All supported Sterling Integrator platforms |
| Related adapters | FTP Client Adapter |
| Application requirements | None |
| Initiates business processes? | Yes, this adapter will initiate another business process in COLLECT mode. |
| Invocation | This adapter can be invoked by an internal service only. |
| Business process context considerations | You must be familiar with the internal service (workflow and business process parameters) that invoked this adapter. Workflow parameters are those values that were passed into the internal service. Business process parameters are those values specified within the business process code. |
| Returned status values | Returned status values:<br>• 0 – Success<br>• 1– Error |
| Restrictions | None |
| Persistence level | System default (Full) |

| Testing considerations | Create a test configuration of the adapter and create a business process to test with. See *Business Process Example*. The most frequent problems encountered are: |
|---|---|
| | • Parameters are configured incorrectly. |
| | • Adapter is not active. |
| | • Response timeout. The IBM server may not always return the status code of the submission before the adapter timeout value is reached. When this happens, the business process is halted and a response timeout message is written to the Advanced Status. This may not mean that the submission has failed, however, because the IBM Server would have processed all the documents and sent them to their respective mailboxes before attempting to send back the response. When writing an On Error script, have the script check for the error message before deciding on whether to retry the submission. Also, set a long timeout value. Or, if possible, check with the trading partner to see if the document has been received. |

## Implementing the IBM IE FTP Adapter

To implement the IBM IE FTP adapter, complete the following tasks:

1. Activate your license for the IBM IE FTP adapter. See *An Overview of Implementing Services*.
2. Create an IBM IE FTP adapter configuration. See *Creating a Service Configuration*.
3. Configure the adapter. See *Configuring the IBM IE FTP Adapter*.
4. Create a business process that includes the IBM IE FTP adapter and enable it.
5. Test the business process and the adapter.
6. Run the business process.

## Configuring the IBM IE FTP Adapter

To configure the IBM IE FTP adapter, you must specify field settings in Sterling Integrator.

## Sterling Integrator Configuration

The following table describes the fields used to configure the IBM IE FTP 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. |

| Field | Description |
|---|---|
| Select a Group | Group of services or adapters of the same type that can act as peers. A Service Group name is used in BPML in place of the Service Configuration name. Service Groups show up in the GPM as if they were Service Configurations. Select a Service Group to associate with this adapter. Valid values are:<br><br>• None – You do not want to include this configuration in a group at this time (default)<br><br>• Create New Group – You can enter a name for a new group in this field, which is then created along with this configuration.<br><br>• 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.<br><br>See *Using Service Groups*. |
| Default IBM IE FTP Server | The host information for IBM IE FTP server. Valid value is a hostname or IP address. Default is myhost. Required. |
| Default IBM IE FTP Server Port | Port number of the IBM IE FTP server. This is typically port number 21, but it can be another number depending on the FTP server settings. Default is 21. Required. |
| User Name | User login name to connect to the IBM IE FTP server. Alphanumeric and case-sensitive. Default is euser. Required. |
| Password | Password used for authentication when connecting to the IBM IE FTP server. Case-sensitive. Default is ******. Required. |
| SMIME Encryption User Certificate | Not in use |
| SMIME Decryption Certificate (System Store): | Not in use |
| SSL | The SSL flag that determines SSL socket negotiation. Valid values are:<br><br>• SSL_IMPLICIT – Indicates that the FTP server expects and requires SSL to happen automatically at the time of connection.<br><br>• SSL_EXPLICIT – Indicates that the FTP client requests SSL and a secure connection is negotiated.<br><br>• SSL_NONE – Indicates that the connection will not use SSL.<br><br>Required. |
| Clear Control Channel | Indicates if information that travels across the control channel should be clear. Valid values are Yes and No. Default is No. Required. |
| Key Certificate Passphrase | Not in use. |
| Cipher Strength | The level of encryption you would like to be applied to the data that flows through the socket connection. Valid values are:<br><br>• ALL<br><br>• WEAK<br><br>• STRONG (default)<br><br>Optional. |

| Field | Description |
|---|---|
| Key Certificate (System Store) | Select from the list of PrivateKeys/Public Certificates that are signed by the trading partner trusted certificate authority. In process data, this parameter is displayed as an object ID. Required for client authentication. |
| CA Certificates | Select from the list of trusted certificate authority public certificates. In process data, this parameter is displayed as an object ID. Required for server authentication. |
| Select the operation | Specifies the type of operation to perform. Valid values are:<br>• COLLECT (default)<br>• SUBMIT<br>Required. |
| Trading Partner | Specifies the trading partner mailbox to collect. Default is account.userid. Required for COLLECT. |
| Message Class | Specifies the message class to collect. Default is text. Required for COLLECT. |
| Business Process | Specifies the business process to invoke. Required for COLLECT. |
| Schedule Settings | Enables you to set up a schedule. Supporting scheduling options are presented based on the value specified here. Valid values are:<br>• Do not use schedule<br>• Run based on timer (default)<br>• Run daily<br>• Run based on day(s) of the week<br>• Run based on day(s) of the month<br>Required for COLLECT. |
| Select the message type | Specifies the message type to send. If you select SUBMIT for the type of operation, you will be presented with this option. Valid values are:<br>• EDI message (default)<br>• Non EDI message<br>Required for SUBMIT. |
| Select the alias table type | Specifies the alias table type. If you select EDI for the message type, this field is displayed. You define the alias table to resolve EDI addresses. Valid values are:<br>• Private alias table – can only be accessed by the user (default).<br>• Global alias table – can be accessed by all users in the system.<br>• Organizational alias table – can be accessed by all users within the same organization.<br>Required for SUBMIT EDI message. |
| Alias table name | Specifies the alias table name. Default is tablename. Required for SUBMIT EDI message. |

## Business Process Example

The following business process example illustrates how you can invoke an IBM IE FTP adapter configuration:

```
<process name="Test_bpml">
  <sequence name="optional">
    <operation name="Invoke an instance of IBM IE FTP adapter ">
      <participant name="IBM_IE_FTP_adapter"/>
       <output message="Xout">
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

# Chapter 110. Instant Messaging Adapter Suite

The Instant Messaging Adapter Suite enables you to use open standard instant messaging protocol, such as XMPP (which works with Jabber® instant messaging), to interact with Sterling Integrator. With the Instant Messaging Adapter Suite, Sterling Integrator can initiate instant messages to alert you of issues needing prompt attention. You can then act on the information, including initiating commands in Sterling Integrator and receiving the results of those commands within the context of the instant message (IM) session.

For the purposes of this documentation, the following terms are defined as follows:

- IM user – User receiving the IM notification initiated by the Instant Messaging Adapter Suite
- Sterling Integrator IM user – IM user account representing Sterling Integrator

The following table provides an overview of the Instant Messaging (IM) adapter suite:

| | |
|---|---|
| System name | The IM Adapter Suite includes services having the following system names:<br><br>• IMBeginSession<br>• IMBeginChat<br>• IMSendMessage<br>• IMReceiveMessage<br>• IMGrammarLoader<br>• IMCommandProcessor<br>• IMEndChat<br>• IMEndSession |
| Graphical Process Modeler (GPM) categories | All Services, Messaging > Instant Messaging<br>**Note:** The IMGrammarLoader service is in the All Services category only. |
| Description | The IM Adapter Suite enables you to carry on an IM session with Sterling Integrator as if the system were another IM user. You can configure multiple Sterling Integrator IM users. Using the IM Adapter Suite, you can send commands to Sterling Integrator, and receive the results of those commands in your IM display. You can pass files to and from Sterling Integrator using IM. To facilitate this, you define all of the commands you will use in a grammar file. The commands in the grammar file correlate to business processes that you create. You can configure the business process running the IM session to loop back after a command executes so that another command can be run. Recommended strategy is to define a command to end the session and make the command part of the IM business process. Sterling Integrator includes a default IM business process, grammar file, and business processes representing the commands in the grammar file. You can edit or copy any of these according to your needs. |

| | |
|---|---|
| Business usage | Enables Sterling Integrator to alert a specified user of a certain event using instant messaging, and enables the alerted user to perform simple queries as to the health and status of the Sterling Integrator system through the IM session. |
| Usage example | You configure the IM Adapter Suite services so that when Sterling Integrator receives an order from a top priority customer, the IM adapter suite will notify the appropriate user by initiating an instant message. The system receives such an order. The IM Adapter Suite notifies the user of the order. |
| Preconfigured? | IM Adapter Suite services have the following service configurations provided with Sterling Integrator, however, some require further configuration to meet your needs, as indicated:<br><br>• IMBeginChat_Instance (requires further configuration)<br>• IMGrammarLoader (requires further configuration)<br>• IMSendMessage_Instance<br>• IMReceiveMessage_Instance<br>• IMEndChat_Instance<br>• IMEndSession_Instance<br>• IMCommandProcessor_Instance<br>• IM_FILE_EXTRACT<br><br>For information about configuring the services, see *Configuring Instant Messaging Adapter Suite Services*. |
| Requires third party files? | Two .jar files are required:<br>• jymsg available at http://jymsg9.sourceforge.net<br>• smack.jar, available from http://www.igniterealtime.org/downloads/<br><br>Use the install3rdparty.sh or install3rdparty.cmd script to install the jar files. For information, see *Implementing the IM Adapter Suite*. To download the XMPP client libraries, go to http://www.igniterealtime.org/downloads/ and download the smack_3_0_4.zip file. For instructions, see *Implementing the IM Adapter Suite*. |
| Platform availability | All supported Sterling Integrator platforms |

| Related services | The IM Adapter Suite includes the following services: |
|---|---|
| | • IM Begin Session – Initiates the IM session as the Sterling Integrator IM user ID. |
| | • IM Begin Chat – Calls the targeted IM user to have the IM session with. |
| | • IM Send Message – Sends messages to the IM user. May send attachments to the IM user. |
| | • IM Receive Message – Receives messages from the IM user. The system uses this service to pause and wait for an interactive message from the user. The receive message service waits for a message until the idle timeout expires. |
| | **Note:** An IMReceiveMessage is not required to handle incoming attachments. Any time a user sends an attachment during an IM session, the IM adapter launches the IM_RECEIVE_FILE business process to handle the attachment. |
| | • IM Grammar Loader (File System adapter) – Loads the IM command syntax from the grammar file. The commands in the grammar file define the commands that the IM Command Processor will recognize. |
| | • IM Command Processor – Receives input from the IM Receive Message service and interprets a received command according to the supplied grammar. When a command matches a grammar file entry, the Processor passes the appropriate business process name to invoke the process. |
| | The value matched from command to grammar file becomes the parameter WFD_NAME in the business process, and a parsed version of the command becomes the primary document. Therefore, input to the IMCommandProcessor is the grammar XML as the primary document, and IMCommand as a process data parameter, and the output of the IMCommandProcessor is the parsed command as the primary document and the matching value as WFD_NAME. |
| | **Note:** Looping capability and business process launching and monitoring take place in the business process. The IM Command Processor only interprets the supplied command. The IMCommandProcessorLoop Business Process handles the looping mechanism and business process execution. |
| | • IM End Chat – Terminates the IM session visual display. |
| | • IM End Session – Terminates the IM session communications. |
| Application requirements | • Create a user account in the target instant messaging application on behalf of the Sterling Integrator system. This is the IM name you provide when configuring the IMBeginSession. |
| | • Establish the user account in the target instant messaging application before running the SendMessage service. |
| Initiates business processes? | No |

| Invocation | To enable command initiation using IM with the IM CommandProcessor, you must: <br> • Write the corresponding business process for the commands <br> • Define the necessary commands in the IM grammar file |
|---|---|
| Business process context considerations | IM Adapter Suite services rely upon session state stored in the business process. For that reason, the services must be used in conjunction with each other. IMBeginSession must always precede calls to IMBeginChat, and IMBeginChat must precede IMSendMessage and IMReceiveMessage. For an illustration, see *Business Process Example 1 – Simple* and *Business Process Example 2 – Complex*. |
| Returned status values | When errors occur, the business processes halt. The status report will specify, with as much detail as available, the failure point. This is vital because there are many process level state variables passed between services that are not manipulated by the user. |
| Restrictions | The system always initiates the IM session. Under no circumstances will Sterling Integrator respond to a session initiated by the IM user. |
| Persistence level | Recommended persistence level is None. |
| Testing considerations | You can create a business process similar to the first example to use for testing. See *Business Process Example 1 – Simple*. |

## Implementing the IM Adapter Suite

To implement the IM Adapter Suite, complete the following tasks:

1. With your IM provider, create a user ID representing Sterling Integrator.
2. Log in to your IM application, and add the ID to your messenger list
3. Log out and then log on as the new Sterling Integrator ID and add your personal user ID or the ID of another IM user to the messenger list of the Sterling Integrator user.
4. Log out of IM.
5. Obtain the **third party XMPP client libraries by downloading the** smack_3_0_4.zip file from http://www.igniterealtime.org/downloads/.
6. Unzip the file to your hard drive and note the location of the files.
7. Download the most recent version of the ymsg_code_jars_v0_6.zip file from http:////jymsg9.sourceforge.net.
8. Unzip the file to your hard drive and note the location of the files.
9. Shut down the application.
10. Unload the following jar files you unzipped in steps 6 and 8:
    • smack.jar
    • ymsg_network_v0_6.jar
    • ymsg_support_v0_6.jar

    To unload the files, use install3rdParty.sh (or install3rdParty.cmd for Windows). The following examples show the commands; you must change the versions and user directories according to your needs:

- `./install3rdParty.sh ymsg_network 0_6 -j` *unzip directory*/
  `ymsg_network_v0_6.jar`
- `./install3rdParty.sh ymsg_support 0_6 -j` *unzip directory*/
  `ymsg_support_v0_6.jar`
- `./install3rdParty.sh smack 3_0 -j` *unzip directory*/`smack.jar`

11. Restart the application.
12. Do you need the adapter to accept to commands from the IM user during
    sessions?
    - If Yes, define the commands in the grammar file. For information, see
      *Default Grammar File* .
    - If No, skip to step 15.
13. In Sterling Integrator, edit the **IMGrammarLoader** configuration of the File
    System adapter. In the **attachFile** field, enter the location and name of the
    grammar file to use. This is the file you modified in step 12.
14. In the GPM, configure the Command Processor service. For information, see
    *Configuring the IM Command Processor Service*.
15. In Sterling Integrator, create a configuration of the IM Begin Session service.
    Enter the IM user ID and password to represent Sterling Integrator in IM
    sessions. For information, see *Configuring the IM Begin Session Service*.
16. In Sterling Integrator or in the GPM, edit the **IM Begin Chat** service
    configuration. Enter the IM user ID to connect with during the session (this is
    your ID or the IM user ID of another user that Sterling Integrator will notify).
    You can enter multiple IDs by separating them with commas. For example,
    mychatid, yourchatid, thirdchatid. For information, see *Configuring the IM
    Begin Chat Service*.
17. Use the IM services in a business process.

## Default Grammar File

The provided grammar file for the IM adapter is an XML-format file that describes
the behavior of the IM Command Processor service. The file, grammar.xml, is
located in the data/xmpp directory of your Sterling Integrator installation
directory.

You can copy and modify this file as necessary in order to define the commands to
be used. This enables the Command Processor to match the correct business
processes to commands received in an IM session. If you use a different grammar
file or change the location of the file, you must modify the IMCommandProcessor
business process to reflect the filename or path change.

The contents of the default grammar file are as follow:

```
<grammar>
<pattern value="get [a-z]* log" result="IM_GET_LOG"/>
<pattern value="show [a-z]*" result="IM_SHOW_COMMAND"/>
<pattern value="bye" result="IM_EXIT"/>
<pattern value="default" result="IM_DEFAULT"/>
</grammar>
```

The grammar file contains two tag types:
- Root – This is always 'grammar.'
- Pattern – A repeatable tag comprised of two required attributes, *value* and *result*.
  This pairing describes the action taken when the system receives a given
  command.

The content of 'value' is a Perl5 type regular expression that will match on command phrases. The content of 'result' is the business process to execute when the value is matched. In the following example, this is illustrated in the first pattern:

```
<pattern value="get [a-z]* log" result="IM_GET_LOG"/>
```

The match will execute the IM_GET_LOG business process when it receives a command of:

```
 get<space><any combination of letters><space>log
```

**Note:** The last pattern in the grammar file, with a value of default, is required in every grammar file. This is the pattern the Command Processor uses when it finds no matching command.

### Example Commands

The following table provides examples of commands that will execute the IM_GET_LOG business process and commands that will not:

| Valid command | Invalid command |
| --- | --- |
| get noapp log | get the big log |
| get all log | get all logs |
| get myspecial log | get log |

### Configuring Instant Messaging Adapter Suite Services

Several of the IM Adapter Suite services require configuration.

### Configuring the IM Begin Session Service

To configure the IM Begin Session service, you must specify field settings in 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:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• Select Group – If you have already created one or more groups for this service type, they display in the list. Select a group from the list. |
| Protocol | IM provider to use for chat sessions. Select **XMPP** or **Yahoo**. Default is **XMPP**. Required.<br>**Note:** Yahoo IM is not supported at this time. |
| Username | IM user ID name to represent Sterling Integrator use during IM sessions. Required. |

| Field | Description |
|---|---|
| Password | IM password for the Sterling Integrator user ID. Optional. |
| Idle Timeout | Length of time in milliseconds to wait for a response from the IM user before halting the process and ending the chat session. Optional. |
| Session Timeout | Length of time in minutes to allow for the entire session to run before halting the process and ending the chat session. Optional. |
| XMPP Server Name | Name of the server of the chat service you are using. Obtain this name from your IM provider. Required. |
| XMPP Server Port | Port number for IM communications. Obtain the port number from your IM provider. Required. **Note:** The default port for XMPP with Jabber is 5222. |
| XMPP Service Name | XMPP service name as defined in the XMPP server if different from the XMPP Server Name value. If omitted, the adapter uses the XMPP Server Name. Optional. |

## Configuring the IM Begin Chat Service

To configure the IM Begin Chat service, you must specify field settings in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| IMChatTo | Name of the IM user for Sterling Integrator to have a chat session with, or a comma delimited list of IM user IDs, for an IM conference. Required. |

## Configuring the IM Send Message Service

To configure the IM Send Message service, you must specify field settings in the GPM:

| Field | Description |
|---|---|
| Config | Name of the service configuration. Required. |
| IMBodyAsAttachment | Valid values are **True** and **False**. When the value is set to True, the system sends the primary document body as an attachment during a SendMessage invocation. When the value is set to False, the system chunks the primary document into multiple messages and sends them to the user. The chunk size is determined by the maximum message size for the IM provider. Use this field only when IMUseBody is set to True. Optional. |
| IMMessage | Message to display on the screen of the IM user. Use this field if you want to specify exactly what the notification will say when Sterling Integrator initiates a chat session. Optional. |
| IMUseBody | Valid values are **True** and **False**. When the value is set to True, the system sends the body of the primary document to the user according to the mode specified in the IMBodyAsAttachment field. Optional. |

## Configuring the IM Grammar Loader Service

To configure the IM Grammar Loader service, you must specify field settings in the GPM:

| Field | Description |
|-------|-------------|
| Config | Name of the service configuration. Required. |
| attachFile | Name of the grammar file for the adapter to reference during the session. Required. |

## Configuring the IM Command Processor Service

To configure the IM Command Processor service, you must specify field settings in the GPM:

| Field | Description |
|-------|-------------|
| Config | Name of the service configuration. Required. |
| IMCommand | Specifies the command for the command processor to match against. Used in conjunction with IMReceiveMessage, this parameter automatically populates according to the message received from the user. Because this is a system-populated value, do not enter a value. Entering a value will force the service to use your supplied value regardless of the actual command received from the IM user. |

## Instant Messaging Adapter Suite Business Processes

The following table describes the predefined business processes associated with the IM Adapter Suite:

| BPML Name | Usage Description |
|-----------|-------------------|
| IM_DEFAULT.bpml | In the event that the adapter receives a command for which there is no match in the grammar file, this is the default process that runs, to indicate that the system does not understand the command. |
| IM_EXIT.bpml | Closes the chat session when the IM user enters an exit command. |
| IM_GET_LOG.bpml | Example business process that returns a Sterling Integrator log file. |
| IM_SHOW_COMMAND.bpml | Lists all commands in the grammar file. |
| IMCommandProcessLoop.bpml | Example process you can configure to run the looping process for proper handling of user-supplied commands within a chat session. |

## Example Business Process Models

This section contains two examples, a simple connect-chat-disconnect business process that you can build and use to test your IM connections, and a more complex example.

## Business Process Example 1 – Simple

Refer to the figure following the steps as you read the process steps. This example business process progresses through IM Adapter Suite services as follows:

1. IM Begin Session service – This service contains the Sterling Integrator IM user ID and password used to log in to IM, and sets basic session parameters. The system logs into IM using the specified ID and password.

2. IM Begin Chat service – This service contains the IM user ID (or comma-delimited list of IDs) the system communicates to when initiating an IM session. The system starts a chat session with the IDs specified.

3. IM End Chat service – When the IM user closes the chat session, this service ends the chat session.

4. IM End Session service – When the chat session has ended, this service signs the Sterling Integrator user ID out of the IM provider application.

The following figure shows the business process in the Graphical Process Modeler:
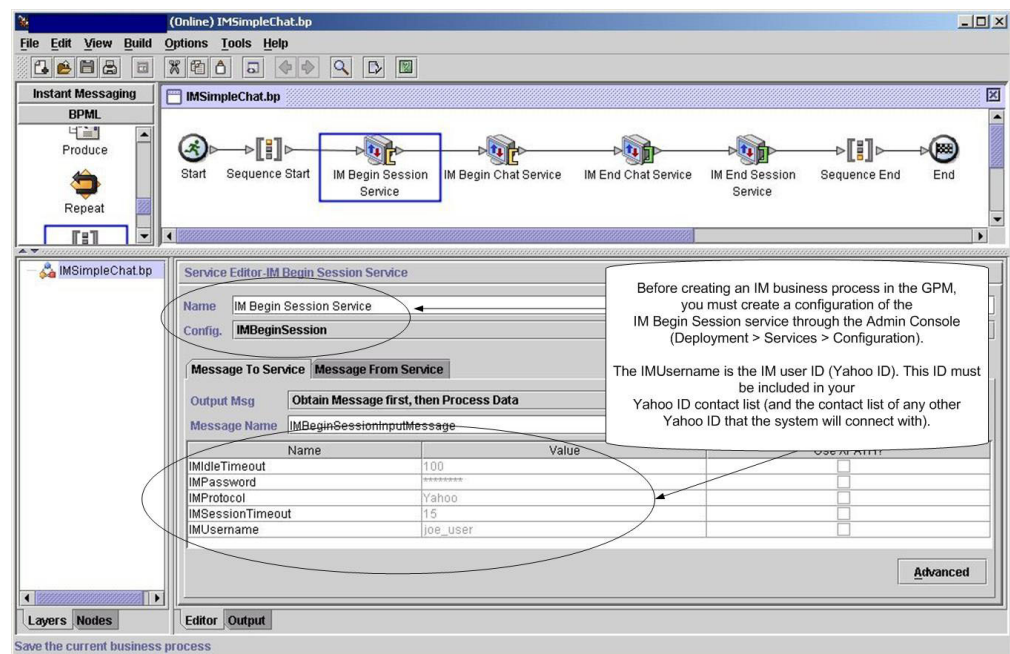


The BPML for the business process would be similar to the following example:

```
<process name="default">
  <sequence>
    <operation name="IM Begin Session Service">
      <participant name="IMBeginSession"/>
      <output message="IMBeginSessionInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="IM Begin Chat Service">
      <participant name="IMBeginChat_Instance"/>
      <output message="IMBeginChatInputMessage">
        <assign to="IMChatTo">myuserid,TomR_Billing,WarehouseUser1</assign>
        <assign to="." from="*"></assign>
      </output>
```

```
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="IM End Chat Service">
      <participant name="IMEndChat_Instance"/>
      <output message="IMEndChatInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="IM End Session Service">
      <participant name="IMEndSession_Instance"/>
      <output message="IMEndSessionInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

## Business Process Example 2 – Complex

This example corresponds with the sample business process
IMCommandProcessLoop.bp provided with your installation of Sterling Integrator.
To use the provided process model and this example, use a configured instance of
the IM Begin Session service to call the process.

Refer to the figures following the steps as you read the process steps. This example
business process progresses through IM Adapter Suite services as follows:

1. The session begins and you see a message in your IM window.
2. You type in a command, which the system evaluates at the decision point in
   the business process model according to the rule, illustrated as the loopRule
   (your process model may use different labels).
3. The the grammar file is loaded and the IM Command Processor service
   attempts to associate the command with a match in the grammar file:
   • If the IM Command Processor service finds a match, it starts the
     corresponding business process. You receive any output of the business
     process in your IM window.
   • If the IM Command Processor service does not find a match, it sends you an
     IM message, saying that the match was not found.
4. The next step depends on the command you sent in step 2:
   • If you sent an end session command, the IM Command Processor ends the
     session and exits the business process.
   • If you sent a command other than to end the session, the IM Command
     Processor restarts the command loop and the business process waits for
     another command.

The following figure shows a similar business process model in the GPM. Because
of the length of the business process, the figure is broken down into three separate
sections here for ease of viewing:

## Section 1



**Testcommandloop.bp**

In the first section of the business process, the system begins a chat session and decides what to do next based on the command received. The rules labeled 'loopRule' determine the activity.

start  Sequence Start  seq1  Assign  IMBeginChat  loop  doLoop  IMReceiveMessa  loopRule  loopRule  done

## Section 2



**Testcommandloop.bp**

doLoop  IMReceiveMessage  File System Adapter  IMCommandProcess or  Invoke Sub-Process  loopBack  Sequence End

loopRule

loop  loopRule  done  Assign  Sequence End  Choice End

In the second section of the business process, the commands entered by the IM user run. For a command to exit the session, the business process uses the bottom loop rule and ends the session. For other commands, the business process continues to run commands and subprocesses according to the top loop rule.

## Section 3



**Testcommandloop.bp**

Sequence End

Choice End  Sequence End  IMSendMessage  IMEndChat  IMCloseSession  Sequence End  End

Sequence End

In the final section of the business process, the IM user's command is to end the session. The system ends the IM chat, logs out of IM, and the business process ends.

# Chapter 111. Import Service

The following table provides an overview of the Import service:

| System name | ImportService |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | This service is used in a business process to automatically import Sterling Integrator resources exported using the Resource Manager, including:<br>• SAP application configurations<br>• Translation maps<br>• Trading partner data (packages, identities, contracts, envelopers, and code lists)<br>• Business processes<br>• Service configurations<br>• XML schemas<br>• XSLT stylesheets<br>• Web templates<br>• Web resources (JSP files, JavaScript files, HTML files, XML files, image files, property files, stylesheets, and custom defined files) |
| Business usage | In a hub and spoke relationship, a hub company could use this service to programatically update information on their trading partners' systems. |
| Usage example | A hub needs to update its trading partner information with all of its spokes. At the same time, it plans on rolling out new XML schemas and translation maps. The hub creates an installable bundle interactively by means of the resource manager. The bundle is sent to the trading partners affected. The bundle is picked up by the trading partners and processed by a business process set up as the updates from trading partner hub x. This process includes the Import service. The service checks the security context and, assuming it is correct, opens the bundle and updates the local system with the updates automatically. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported application platforms |
| Related services | This service is designed to work in conjunction with a transport type service. The transport service brings the resources into the local system. |
| Application requirements | Before using this service, a security context for the installable resource bundle must have been created using the Security Context utility. |
| Initiates business processes? | No |
| Invocation | Event driven. |

| Business process context considerations | The Import service configuration may contain context and identity values for a security context, which are used during the business process execution to fetch the passphrase for verification if the file being imported contains encrypted data. If the passphrase is required but the values are not for the correct passphrase or no security context information is available (either the values from the service configuration or the passphrase value stored for each configuration in the Sterling Integrator database), the Import service will fail. |
|---|---|
| Returned status values | Returned status values:<br><br>• Success – The service completed successfully.<br><br>• Error – The service experienced a fatal error while processing. |
| Restrictions | • This service does not construct installable bundles or export resources. These operations must be done interactively.<br><br>• All resources defined in the installable bundle will be installed.<br><br>• Any existing resources will be updated, and the version number incremented.<br><br>• The installed resource will become the default, if applicable. |
| Persistence level | Full |
| Testing considerations | Export a set of resources from Sterling Integrator to a file called Export.xml. Import these resources in to another Sterling Integrator server. Check the status report. There must not be any errors and it should be possible to test the imported resources. |

## How the Import Service Works

The Import service exercises the same functionality as the Import Resources option in the Resource Manager, with one exception—the service has no user interaction, so the service does not ask for confirmation of options. When using the Import service, all available resources are imported, and all imported versions are set as the defaults, where applicable.

You create a security context for an installable bundle, which can prevent unauthorized users from creating or updating resources.

The Import service works with the Security Context utility in the application. The utility is called securityContext.sh (for Unix) or securityContext.cmd (for Windows). It is located in the bin directory of your Sterling Integrator installation.

This is an example of how the security context is used:

1. A developer at company A exports a resource bundle to be sent to company B, where the bundle will be imported. If required for the type of resource to be exported, the developer creates a passphrase for the resource bundle as a part of the export process.

2. After the export is complete, the developer sends the resource bundle to the company B system administrator, and also informs the system administrator of the passphrase.

3. The system administrator at company B uses the Security Context utility to enter the passphrase into the Sterling Integrator database and to create a security context.

4. The system administrator passes the resource bundle and the name of the security context to a developer.

5. The developer configures the Import service in the GPM, using the context and identity values from the security context that the system administrator provided.

For more information about creating a security context, see *Using the Security Context Utility*.

## Implementing the Import Service

To implement the Import service, complete the following tasks:

1. After receiving a resource bundle from a trading partner, create a security context for it. For information, see *Using the Security Context Utility*.

2. Create an Import service configuration. See *Managing Services and Adapters*.

3. Configure the Import service.

4. Use the Import service in a business process.

## Configuring the Import Service Configuration

To configure the Import service, you must you must specify settings for the following fields in Sterling Integrator:

| Field | Description |
|---|---|
| Backup | Identifies the path where the backup is saved. If the path is invalid during backup, the file is written to <install>/tmp and a message is added to the Import Report indicating the location. If the parameter is not specified, then the backup is not generated. |
| Config | Name of the service configuration. |
| Context | The company from which the resource files are obtained. Required if the file to be imported contains encrypted data; otherwise, optional. Example: Company_*x* |
| Identity | An ID to identify various passphrases received on various dates from the same company. Required if the file to be imported contains encrypted data; otherwise, optional. Example: 10Jan2004 |
| KeepExistingControlNumbers | Specifies whether the control numbers in the import file will be imported. The default for this parameter is **No**, which specifies that the control numbers in the import file will be imported. If you change this parameter to **Yes**, it specifies that for existing envelopes and control numbers, control number values in the import file will not be imported. If a version of an envelope or control number being imported already exists in the system, the import process will overwrite the value specified in the import file with the existing control number value for that envelope or control number. |

## Output from Business Process to Service

The following table describes the output from the business process to the Import service:

| Parameter | Description |
|---|---|
| Filename | The name of the resource file, including full path information. Valid value is any valid path and filename. |

## Using the Security Context Utility

There are three actions you can perform with the security context command: list, get, and set. The security context command file is located in the bin directory of your Sterling Integrator installation.

| Action | Description | Usage |
|---|---|---|
| list | Lists all security contexts available. | UNIX: *install_dir/* bin>securityContext.sh list_context<br><br>Windows: *installdir\\* bin>securityContext.cmd list_context |
| set | Updates the database in Sterling Integrator with the new context. Takes three parameters:<br><br>• context<br>• identity<br>• passphrase | UNIX: *install_dir/* bin>securityContext.sh set *context identity passphrase*<br><br>Windows: *install_dir\\* bin>securityContext.cmd set *context identity passphrase*<br><br>Returns the following message: Context saved. |
| get | Returns the passphrase value for the context. Takes two parameters: context and identity. | UNIX: *install_dir//* bin>securityContext.sh get *context identity*<br><br>Windows:<br><br>*install_dir\\*bin>securityContext.cmd get *context identity*<br><br>Returns the following values: *context*, *identity*, *password* |

## Example

In the following example, the Kimata company's system administrator, Jill, creates a security context called MaxxMart for an exported resource bundle just received from their trading partner, MaxxMart. Jill sets the identity for this context to dec19 (date it was received from the trading partner). MaxxMart also sent Jill the passphrase that they created for the resource bundle: bubblegum.

```
install_dir\bin>securityContext.sh set MaxxMart dec19 bubblegum
```

In the second example, Jill wants to find out what security contexts are on her Sterling Integrator system, and uses the list_context action to find out. There are three contexts on the system: MaxxMart, Taylor, and Zapf.

```
install_dir\bin>securityContext.sh list_context
Contexts:
MaxxMart,Taylor,Zapf
```

In the third example, Jill wants to know what the passphrase is for the security context named Taylor that has an identity of jan20. She uses the get action and finds that the passphrase is thunder.

```
install_dir\bin>securityContext.sh get Taylor jan20
Taylor,jan20,thunder
```

## Business Process Example 1

The following example illustrates using the Import service in a business process to import a resource file called dec19 from Company_x:

```
<process name="ImportService">
  <sequence>
    <operation>
      <participant name="ImportService"/>
      <output message="Xout">
         <assign to="Context">company_x</assign>
         <assign to="Identity">dec19</assign>
         <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
         <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

## Business Process Example 2

The following example illustrates using the Import service in a business process to import a resource file called april1 from RomansFloorsAndMore, using the Keep Existing Control Numbers option (specifying that for existing envelopes and control numbers, control number values in the import file will not be imported, and i)f a version of an envelope or control number being imported already exists in the system, the import process will overwrite the value specified in the import file with the existing control number value for that envelope or control number:

```
<process name="ImportServiceWithKeepExistingControlNumbers">
<sequence>
    <operation>
      <participant name="ImportService"/>
      <output message="Xout">
 <assign     to="Context">RomansFloorsAndMore</assign>
 <assign to="Identity">april1</assign>
 <assign to="KeepExistingControlNumbers">True</assign>
 <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
         <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

## Viewing the Import Service Status Report

Once you have imported resources using the Import service in a business process, it is a good idea to check that all resources were imported successfully. Also, you

may need to use the report for troubleshooting if the service and business process fail. You can view the status report from the Business Process Detail page.

To view the report Business Process Detail page, complete the following steps:
1. From the Business Process menu, select **Monitor** > **Current Processes**. Current business processes are displayed in a list.
2. Click the Instance ID next to the desired business process. The Business Process Detail page for that business process displays.
3. In the Status Report column, click the Info icon for the Import service. The report is opened in another window. The status of each resource that the service attempted to import is shown, which enables you to verify whether each was successfully imported or not.

Some reasons that the Import service might fail (which will cause the business process to fail) in situations where passphrase (Context/Identity) is required are:
- Invalid passphrase (the passphrase in the database doesn't match the passphrase in the resource bundle).
- No passphrase in situation where passphrase is required (possibly there was no security context created for this resource bundle).
- Either the context or identity value in the Import service configuration used in the business process is wrong or was left blank.

The following is a sample status report for an import bundle. There is one error for a resource that could not be imported (transport account password):

```
Name: UpdateTPInfo      Instance ID:1053     Service Name:
  Import Service Status report on 2004-03-12 14:45:19.16 for service:
  Import Packaging :: packaging_1079119091618 :: update :: SUCCESS ::
Resource successfully imported.
Identity :: MaxxMart :: update :: SUCCESS :: Resource successfully imported.
Transport :: HTTP Transport :: create :: Message :: Error decrypting
transport
account password...value will be stored as it was in import file.
Transport :: HTTP Transport :: update :: SUCCESS :: Resource successfully
imported.
Document Exchange :: MaxxMart Doc
Exchange :: update :: SUCCESS :: Resource successfully imported.
Delivery Channel :: ABCD :: update :: SUCCESS :: Resource successfully imported.
Profile :: MaxxMart1 :: update :: SUCCESS :: Resource successfully imported.
End of report
ImportService stayed in queue 7  ms
```

# Chapter 112. Index Business Process Service

The Index Business Process service sets all completed or terminated business process data to be ready for archiving or purging. Creates an entry for each completed or terminated business process in the WF_INST_S table, and updates several tables. The following table provides an overview of the Index Business Process service:

| System name | Index_Service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, System |
| Description | Sets all completed or terminated business process data to be ready for archiving or purging. Creates an entry for each completed or terminated business process in the WF_INST_S table, and updates these tables: <ul><li>WORKFLOW_CONTEXT</li><li>DOCUMENT</li><li>DOCUMENT_EXTENSION</li><li>DATA_TABLE</li><li>CORRELATION_SET</li><li>WORKFLOW_LIFESPAN</li><li>DOCUMENT_LIFESPAN</li><li>WORKFLOW_DATA</li></ul> |
| Business usage | This service selects all complete and terminated business processes and flags them for archive or purge. |
| Usage example | Sterling Integrator runs a business process with the Index Business Process service information. BMPL can run manually or by schedule. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms. |
| Related services | Archive Business Process service, Purge service. |
| Application requirements | No |
| Initiates business processes? | No |
| Invocation | Runs by only the internal scheduler. |
| Business process context considerations | No |
| Returned status values | The following status values are returned after this service runs in a business process: <ul><li>100s – Workflow engine error, for example, the service is deactivated</li><li>200s – System error, for example database error</li><li>300s – The service configuration error.</li></ul> |
| Restrictions | There can be only one configuration per server of this service. |

## Implementing the Index Business Process Service

To implement the Index Business Process service, complete the following tasks:

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

## Configuring the Index Business Process Service

To configure the Index Business Process service, you must specify settings for the following fields in 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:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. |
| Run As User | Enter the user ID to associate with this service when run. |
| Do not use schedule | If field is selected, it does not get scheduled to run. |
| Run service based on timer every | Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. |
| Run service daily at | Valid values are the hour and minutes at which to run the service, daily. Indicate whether you want the service to run at startup. |
| Run service weekly on | Valid values are the day of the week, the hour, and the minutes at which to run the service. Indicate whether you want the service to run at startup. |

# Chapter 113. Introduction

This guide contains documentation for services and adapters provided with Sterling Integrator 5.0 only.

For information about services and adapters provided with related products such as Sterling Standards Library or Sterling e-Invoice Gateway, see the respective section of the Sterling Integrator 5.0 Documentation in PDF Format page http://www.sterlingcommerce.com/Documentation/SI50/PDFs.htm.

# Chapter 114. Invoke Sub-Process Service

> **Note:** This service is also referred to as the Invoke Business Process service and the Invoke service.

The following table provides an overview of the Invoke Sub-Process service:

| System name | Invoke Sub-Process Service |
|---|---|
| Graphical Process Modeler (GPM) category | All Services, Sync Mode, Transactional Mode |
| Description | Starts a subprocess. |
| Business usage | The Invoke Sub-Process service provides a way to reuse the same business process in multiple business processes by enabling you to invoke the subprocess from within a parent process. |
| Usage example | Runs a standard deenveloping (for example, EDI or SOAP) business process to extract business payload from a document bundle for transmission. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | No |
| Application requirements | No |
| Initiates business processes? | Yes. Each call to the Invoke Sub-Process service starts one subprocess (child). The Invoke Sub-Process service can be called multiple times from a business process. Subprocesses can be started in asynchronous or synchronous mode. Subprocesses can optionally be run as separate processes, or inline as part of the parent business process. |
| Invocation | If a child process is disabled, a service is disabled, or there is a license error causing the subprocess to halt, the parent will not continue. Additionally, if the subprocess is manually stopped, or stopped using a system shutdown, the parent will not continue.<br>**Note:** You can fix the failures to continue the parent by resuming the subprocess. |
| Business process context considerations | The Invoke Sub-Process service adds the business process instance ID to the INVOKE_ID_LIST object stored in the business process context. The Invoke Sub-Process service also places the business process instance ID of the generated sub-process into the IWF_Id name/value pair and stores it in the business process context. The tracking feature uses this value to be able to show information about subprocesses. This value is then deleted from the business process context before any services that follow are run. |

| Returned status values | Returned status values: |
|---|---|
| | • Success – If the subprocess was started correctly (async, inline modes) or if the child executed successfully (sync or inline modes). |
| | • Error – If the subprocess was not started correctly (async) or if the child did not execute successfully (all modes). |
| | When running in embedded mode: |
| | • Error – From the error step in the subprocess, if the subprocess ended with an error. |
| | • Success – if the subprocess completed. |
| Restrictions | Restrictions by mode are: |
| | • Synchronous |
| | When the Invoke Sub-Process service is set to synchronous mode, the parent business process suspends processing until it receives data from the child business process. In synchronous mode, the parent is notified when the child encounters errors. |
| | • Asynchronous |
| | When the Invoke Sub-Process service is set to asynchronous mode, the parent and child process simultaneously and independently of each other. Therefore, the parent does not receive notification when the child encounters errors. |
| | • Inline |
| | When the Invoke Sub-Process service is set to run a subprocess inline, the subprocess runs as part of the parent process, sharing the same process data. |
| | • Embedded |
| | Use to run a subprocess with no persistence. |
| | **Note:** A subprocess run inline will be displayed in tracking just like other steps in the parent process. |
| Persistence level | Full (except in Embedded mode, which runs the subprocess with no persistence)<br>**Note:** When persistance level is error only, sync invoke mode is not supported. |
| Testing considerations | None |

## How the Invoke Sub-Process Service Works

When the Invoke Sub-Process service is set to synchronous mode, the parent suspends processing until it receives data from the child. In synchronous mode, the parent is notified when the child encounters errors.

When the Invoke Sub-Process service is set to asynchronous mode, the parent and child process data simultaneously and independently of each other. Therefore, the parent does not receive notification when the child encounters errors.

When the Invoke Sub-Process service is set to run a subprocess inline, the subprocess runs as part of the parent process, sharing the same process data.

When the Invoke Sub-Process service is set to run in embedded mode, the subprocess runs with no persistence, meaning that no record of the process is

recorded in Sterling Integrator and no tracking is done.

## Performance Tips

By default, when you use the Invoke Sub-Process service in a business process, all process data passes from the parent process to its subprocess.

However, if you are using the Invoke Sub-Process service in sync mode, a special tag called 'message_to_child/message_to_parent' enables you to pass along only the 'message_to_child/message_to_parent' node in the process data of the parent process or subprocess. Using this tag can provide significant performance improvement.

Before invoking a subprocess, create a special tag called 'message_to_child' in the parent process, and append all of the data needed in the subprocess under this node. The Invoke Sub-Process service will pass only this node to the subprocess. The following example shows how the BPML might look for this:

```
<process name="MessageToChildInvokeDoc">
  <sequence name="simple">
    <assign to="Msg1" from="'Hello'" append="true"/>
    <assign to="Msg2" from="Msg1" append="true"/>
    <assign to="Msg3" from="Msg1"/>
     <assign to="OtherMsg4" from="//Msg2" append="true"/>
    <assign to="OtherMsg5/OtherMsg6" from="//Msg2/Msg1/text()" append="true"/>
    <assign to="message_to_child" from="//PrimaryDocument | // OtherMsg5 " append="true"/>
  <operation>
     <participant name="InvokeBusinessProcessService"/>
       <output message="Xout">
         <assign to="." from="*"></assign>
         <assign to="WFD_NAME">MessageToParentNode</assign>
         <assign to="INVOKE_MODE">SYNC</assign>
       </output>
       <input message="Xin" >
          <assign to="." from="*"></assign>
       </input>
    </operation>
 </sequence>
</process>
```

Or, you can specify the return node from the input message in the Invoke Sub-Process service. That is, only the ChildDoc node will be returned from the subprocess. In this case, you do not need to specify the 'message_to_parent' node in the subprocess. This is shown in the following example:

```
<operation>
     <participant name="InvokeBusinessProcessService"/>
       <output message="Xout">
         <assign to="." from="*"></assign>
         <assign to="WFD_NAME">MessageToParentNode</assign>
         <assign to="INVOKE_MODE">SYNC</assign>
       </output>
       <input message="Xin" >
          <assign to="test1" from="//ChildDoc"></assign>
--- this 'ChildDoc' node is the sub node of the process data  in its
sub process MessageToParentNode---------------------------
       </input>
    </operation>
```

If you are not specifying the return node in input message in the Invoke Sub-Process service, you can, in the subprocess, create a tag 'message_to_parent' and append all of the data needed from the subprocess to the parent process under this node. Only data under this node will be returned to the parent process. The default is passing all process data to the parent process.

Example:

```
<process name="MessageToParentNode">
  <sequence name="simple">
    <assign to="ChildMsg1" from="'Hello World'" append="true"/>
    <assign to="ChildMsg2" from="ChildMsg1" append="true"/>
    <assign to="ChildMsg3" from="ChildMsg1"/>
    <assign to="ChildDoc" from="PrimaryDocument/@SCIObjectID" append="true"/>
    <operation name="Set Document">
            <participant name="XMLEncoder"/>
            <output message="XMLEncoderTypeInputMessage">
            <assign to="mode">process_data_to_document</assign>
            <assign to="root_element">DocumentSub</assign>
            <assign to="xPath">/ProcessData</assign>
            <assign to="." from="*"/>
            </output>
            <input message="inmsg">
            <assign to="xmldoc" from="PrimaryDocument" append="true"/>
            </input>
    </operation>
    <assign to="message_to_parent" from="//PrimaryDocument | //ChildMsg3" />
  </sequence>
</process>
```

## Implementing the Invoke Sub-Process Service

To implement the Invoke Sub-Process service for use in a business process, complete the following tasks:

1. Create a configuration of the Invoke Sub-Process service. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Invoke Sub-Process Service*.

2. Specify field settings for the service configuration in the Sterling Integrator Admin Console and in the GPM as necessary. For information, see *Configuring the Invoke Sub-Process Service*.

3. Use the Invoke Sub-Process service in a business process.

## Configuring the Invoke Sub-Process Service

To configure the Invoke Sub-Process service, you must specify settings for the following fields in the GPM.

**Note:** Any field values passed from a prior service can override any of configured fields for this service.

| Field | Description |
|-------|-------------|
| Config | Name of the service configuration. Required. |
| WFD_NAME | Business process used in the service configuration. Valid value is the name of any business process checked into the system. Required. |

| Field | Description |
|---|---|
| INVOKE_MODE | Mode in which to run the subprocess. Optional. Valid values are:<br><br>• async = asynchronous (default)<br>• sync = synchronous<br>• inline = In-line<br>• embedded = starts business process in SYNC mode with Enable Transaction on, and runs the subprocess in the same transaction as the Invoke Sub-Process service. The subprocess is run with no persistence.<br><br>**Note:** When persistance level is error only, sync invoke mode is not supported.The following optional parameters can be used with embedded mode:<br><br>• START_WITH_DOC – Use if the subprocess needs to be started with a document. The process checks the DOC_URL parameter for a filename (see following parameter) first. If the DOC_URL parameter is set to NONE, then the primary document is passed to the subprocess. This parameter is required if the subprocess needs a document from the parent business process.<br>• DOC_URL – Name of the document from which the subprocess will retrieve data. Use this parameter with the START_WITH_DOC parameter if the parent primary document is not the document needed to start the subprocess.<br>• DOC_ENCODING – Use if the document that is used to start the subprocess require different encoding from the default encoding.<br>• USER_NAME – Use if the subprocess requires a different user name to start (permission to start).<br>• PASS_DOC – Use if the subprocess needs to pass its primary document to back to the Invoke Sub-Process service. |
| PARM_LIST | List of business process parameters to override. Valid values are:<br><br>• Name1=value1<br>• &name2=value2<br>• &name3=value3<br><br>Optional. |

| Field | Description |
|---|---|
| NOTIFY_PARENT_ON_ERROR | Errors the subprocess reports to the parent business process. This parameter is valid for synchronous mode only. Valid values are:<br><br>• NOTIFY_PARENT_ON_ERROR_SERVICE_ ERRORS_ONLY = SERVICE - Notify parent of the error only if a service-generated error occurs in the subprocess.<br><br>• NOTIFY_PARENT_ON_ERROR_ ALL = ALL - Notify parent of the error if any type of error occurs in the subprocess.<br><br>• NOTIFY_PARENT_ON_ERROR_ NONE =NONE - This option does not notify the parent of the error from the subprocess. The subprocess completes with status error.<br><br>• NOTIFY_PARENT_ON_ERROR_SYSTEM_ ERRORS_ONLY = SYSTEM - Notify parent of the error only if a system-generated error occurs in the subprocess.<br><br>Default is NOTIFY_PARENT_ON_ERROR_ALL = ALL. Optional. |
| PASS_STATUS_REPORT | Pass status report to parent. Valid values are:<br>• ON_SUCCESS<br>• ON_ERROR<br>• ALWAYS<br>• NEVER<br><br>Default is ON_ERROR. Required. |
| COPY_SERVICE_PARMS | Invoke service parameters passed to the subprocess. Valid values are True and False. Default is True. Required. |

## Parameters That Must Be Added in BPML

The following additional parameter is available for use with the Invoke Sub-Process service, but can only be added by editing your business process manually. This parameter is not available through the Admin console or the GPM:

| Parameter | Description |
|---|---|
| SAME_TRANSACTION | Use this parameter to allow the Invoke business process service and the first step of the sub-business process to run in the same transaction.<br><br>If set to true, the first step of the sub-business process runs in the same transaction as the Invoke business process service. If either the Invoke business process service or the first step of the sub-business process, the sub-business process fails to start and the Invoke business process service has an error status.<br><br>If set to false, the first step of the sub-business process runs as an independent transaction and is not part of the Invoke business process service transaction. The sub-business process may be in an active state or error state.<br><br>SAME_TRANSACTION should be manually set to false. The default is true.<br><br>When the sub-business process definition has the following setting, SAME_TRANSACTION will not be valid from the Invoke business process service. The system will internally set it to false:<br>• startMode =Sync<br>• Transaction =TRUE<br>• persistence_level=PERSISTENCE_WF_NONE,<br>• persistence_level=PERSISTENCE_ERROR_ONLY |
| DOC_STREAM | Use this parameter to allow the sub-business process to utilize document streaming to minimize the memory footprint for documents passed to it from the parent process.<br><br>This is useful for large documents that could potentially result in extreme memory usage.<br><br>This parameter is valid only when the Invoke Sub-Process service is set to run in embedded mode. |

## Business Process Example

The following BPML example illustrates using the SAME_TRANSACTION parameter:

```
<operation>
        <participant name="InvokeBusinessProcessService"/>
        <output message="Xout" >
        <assign to="INVOKE_MODE">SYNC</assign>
        <assign to="WFD_NAME">sleepy.bpml</assign>
        <assign to="SAME_TRANSACTION">false</assign>
        </output>
        <input message="Xin" >
        <assign to="." from="*"></assign>
        </input>
</operation>
```

## Output from Service to Business Process

The following table describes the output from the Invoke Sub-Process service to the business process:

| Parameter | Description |
|---|---|
| INVOKE_ID_LIST | A generated list of business process IDs for the processes running. |
| WFD_VERSION | Version of the business process captured by the system at run time. This is a system assigned parameter and is not configurable. |

## Business Process Example

The following example illustrates how the Invoke Sub-Process service could be used in a business process with Async mode:

```
<operation>
      <participant name="InvokeBusinessProcessService" />
            <output message="Xout">
                    <assign to="INVOKE_MODE">ASYNC</assign>
                    <assign to="WFD_NAME">mybusprocess</assign>
                    <assign to="." from="*"></assign>
            </output>
            <input message="Xin" >
                    <assign to="." from="*"></assign>
            </input>
</operation>
```

## Running a Business Process with No Persistence

To enable a business process to run with no persistence, you must start it as a subprocess (child) of another business process (parent). Include the Invoke Sub-Process service in the parent business process, and set the service to use Embedded mode. Embedded mode starts the business process in SYNC mode with Enable Transaction on, and runs the subprocess in the same transaction as the Invoke Sub-Process service. The subprocess is run with no persistence.

When checking in the parent process to Sterling Integrator, on the Process levels page, make sure that:
- Enable Async Start Mode Select is not selected (so that the business process starts in SYNC mode)
- Enable Transaction is selected

The following example illustrates using the Embedded mode:

```
<operation>
<participant name="InvokeBusinessProcessService" />
<output message="Xout">
<assign to="INVOKE_MODE">EMBEDDED</assign>
<assign to="WFD_NAME">mybusprocess</assign>
<assign to="." from="*"></assign>
</output>
<input message="Xin" >
<assign to="." from="*"></assign>
</input>
</operation>
```

# Chapter 115. iWay Adapter

The following table provides an overview of the iWay adapter:

| System name | iWayadapter |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services, Applications > ERP |
| Description | Enables Sterling Integrator to communicate with over 100 adapters that iWay communicates with through the iWay XML Transformation Engine (iXTE) server. Adapter types supported by the iWay adapter include application systems adapters (ERP and CRM) and data adapters (relational data and connector technologies). |
| Business usage | Sends documents from backend systems through Sterling Integrator to iWay. |
| Usage example | You are sending data to a PeopleSoft® application through iWay. The data is sent from your system to Sterling Integrator, which translates the data into an XML document. The business process runs the iWay adapter, which takes the data, envelopes it, and sends it to iWay. |
| Preconfigured? | No |
| Requires third party files? | Yes. The iWay Application Explorer version 5.2.1 or higher is needed to create XML schemas that can be used in a Sterling Integrator translation map. |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | XML Translation service |
| Application requirements | Sterling Integrator supports iWay version 5.2.1 and higher. To use the iWay adapter, you must have:<br>• iWay software installed and configured<br>• iWay documentation<br>• The appropriate adapter documentation |
| Initiates business processes? | Yes |
| Invocation | Runs as part of a business process. |
| Business process context considerations | Resultant workflow context contains response from iWay adapter. It does not contain the original request. In bootstrap mode, a workflow context is created by the iWay adapter. |
| Returned status values | Success, failureA status report is generated on failure. |
| Restrictions | None |
| Persistence level | System default (Full) |
| Testing considerations | To test outbound, an iWay instance must be running and a listener configured on the iWay server. To test inbound, the iWay adapter agent must be installed on the iWay server. |

## Requirements

Sterling Integrator supports iWay version 5.2.1 and higher. To use the iWay adapter, you must have:

- iWay software installed and configured
- iWay documentation
- The appropriate adapter documentation

For information about iWay software, see http://www.iWaysoftware.com.

## Before Using the iWay Adapter

Before you can use the iWay adapter, you must create XML schemas and translation maps to generate XML documents that can be sent to iWay. For each iWay configuration you have, use the iWay Application Explorer to create two XML schemas: one for Request documents and one for Response documents. For example, you may have separate iWay configurations for PeopleSoft, UCCnet, and SAP.

Follow these steps:

1. Determine how many iWay configurations you will integrate with Sterling Integrator by using the iWay adapter.
2. Use the iWay Application Explorer to create Request and Response XML schemas for each.
3. Check in the XML schemas to Sterling Integrator.
4. Using the Sterling Integrator Map Editor, create maps for each schema to translate from your document format to the format required for iWay.
5. Test the maps to verify that no translation errors exist.

For more information about the iWay Application Explorer, see your iWay documentation or the iWay website: http://www.iWaysoftware.com.

## How the iWay Adapter Works

The following steps summarize how the iWay adapter works in an Sterling Integrator business process:

1. The business process passes XML data containing the executable command to the iWay adapter.
2. The iWay adapter envelopes the incoming data into an iWay RequestXML document, and sends this request to the iWay Software iXTE server.
3. The iWay adapter waits for a response from the iWay Software iXTE server.
4. The iWay Software iXTE server receives the RequestXML document and passes it on to the appropriate iWay Software Intelligent adapter using a preconfigured datasource.
5. The iWay Software Intelligent adapter executes the request against the back-end system and returns an answer set to the iWay Software iXTE server.
6. The iWay Software iXTE server passes the results back to the Sterling Integrator iWay adapter in the form of a ResponseXML document.
7. The iWay adapter parses the response (ResponseXML) and places the data into the output business process context. The status of the output business process context is SUCCESS and the primary document contains the result set.

**Note:** Not all commands produce a result set.

Note the following exceptions:

- If the XML data produces an empty result set (as with delete or update requests), the status of the output business process context is SUCCESS and the primary document contains the empty result set.
- If the returned ResponseXML document contains an error, the ResponseXML document is parsed and the data is placed into the output business process context. The status of the output business process context is ERROR and the primary document contains the RequestXML document data.
- If the returned ResponseXML document is not received within the specified time (default 30 seconds), the status of the output business process context is ERROR and the primary document contains the Request XML document data.

8. The business process continues with the next activity.

## Implementing the iWay Adapter

To implement the iWay adapter, complete the following tasks:

1. Activate your license for the iWay adapter. See *An Overview of Implementing Services*.
2. Create an iWay adapter configuration. See *Creating a Service Configuration*.
3. Configure the iWay adapter. See *Configuring the iWay Adapter*.
4. Create and enable a business process that includes the iWay adapter configuration.

   **Note:** The business process must be set up to move data between Sterling Integrator and one or more of the systems supported by the iWay adapter.
5. Test the business process and the adapter.
6. Run the business process.

## Configuring the iWay Adapter

To configure the iWay adapter, you must specify field settings in 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, 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:<br><br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** See *Using Service Groups*. |
| iWay hostname(HostName) | Host name or IP address of the iWay iXTE server. Required. |
| iWay port number(PortNumber) | Port number of the iWay iXTE server listener. Required. |
| Inbound Service Listen Port(listenPort) | Enter the port number for the iWay adapter to listen on. Required. |
| Inbound Service Bind Address(listenHostname) | Host name and port number for the iWay adapter to listen on. Optional.<br>**Note:** This parameter provides multi-homed server support and allows tighter security. By default, the iWay adapter will bind to all available homes (network interfaces) on the iWay iXTE server and listen on all of them. If a host is entered for this parameter, the iWay adapter will only listen on that host. If left blank, the adapter will use the default behavior and listen on all available homes. |
| Bootstrap Process(bpID) | If using the adapter to initiate a business process, select the name of the business process from the list. |
| Document Storage Type(docStorageType) | Defines how the document will be stored in the system. RequiredValid values:<br><br>• System Default<br>• Database<br>• File System<br><br>**Note:** See *Selecting a Document Storage Method for Bootstrap Adapters*. |

## Deploying the GIS iWay Agent

To deploy the GIS iWay Agent on the iWay iXTE server:

1. Copy the *GIS_Installation*/client/iWay/GISiWayAgent.jar file to the system that contains the iXTE server.
2. From the iXTE console, select **Configuration** > **Register Libraries**
3. Add the GISiWayAgent.jar file from this interface.
4. Select **Configuration** > **Defines** > **Agent**
5. In the New Agent dialog box, enter the following:
   - For **Alias,** enter `GISAgent`
   - For **Procedure,** enter `com.sterlingcommerce.woodstock.services.iWay.agent.GISRequestAgent` (*gisservername,iWay adapter Listen port,GIS User,GIS Password*)

For example, assume Sterling Integrator is running on a machine named **fred** and the iWay adapter is configured to listen on port 50000 and you have a user called `joe` with a password of `wilma`. The resulting procedure setting would be:

```
com.sterlingcommerce.woodstock.services.iWay.agent.GISRequestAgent
(fred,50000,joe,wilma)
```

- For **Comment**, enter `GIS Request Agent`

6. Configure your listener to use this agent:

   a. Go to **Configuration/Listeners** and select the listener to configure.

   b. In the upper right hand corner, click **Agents**.

   c. Select **Add Agents** > **GISAgent**.

   d. Click **Add**.

Anything received by this listener will now be forwarded to Sterling Integrator.

**Note:** The Sterling Integrator iWay Agent only works with XML payloads.

# Chapter 116. Java Task Service

The Java Task service executes java code in Sterling Integrator. It eliminates the need to create a custom service. The following table provides an overview of the Java Task service:

| System Name | JavaTask |
|---|---|
| Graphical Process Modeler (GPM) categories) | All Services |
| Description | Executes java code in Sterling Integrator. |
| Business usage | Used to execute java code without having to create a custom service. |
| Usage example | To execute any piece of javacode within Sterling Integrator. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Application platforms. |
| Related services | No |
| Initiates business processes? | A business process can be initiated if the code is written to initiate it. |
| Invocation | None |
| Business process context considerations | None. |
| Returned status values | Return of a string. |
| Restrictions | None |
| Persistence level | Defaults to Full. |
| Testing considerations | None |

## How the Java Task Service Works

To execute the source code place import statements at the top followed by any set of standard java code you want to call. There are two classes automatically available to you, one called wfc (which is an instance of the workflow context of the current process), and one called log (which is an instance of the XLogger). There must be a return of a string included in the code fragment. This string will go to the log.

Example:

```
import com.sterlingcommerce.woodstock.workflow.Document;Document
doc = wfc.getPrimaryDocument();if (doc == null){ log.log("Document
is null");}else{ log.log("Document Name:" + doc.getDocumentName());}return
"return value here";Implementing the Java Task Service
```

To implement the Java Task service, complete the following tasks:

1. Create a configuration of the Java Task service.
2. Create a business process with this service in the GPM.
3. Specify the Java source field with either Java code or the relative path to a file containing Java code. Use the settings in Configuring the Java Task Service for more information.

4. Execute the business process.

## Configuring the Java Task Service

The task takes two parameters, javaSrc and srcLocationMode. The javaSrc can either be the full path to a file containing the java code you wish to execute, or it can be the actual java code you wish to execute. The srcLocationMode will either be "File" if you are supplying a filename and "Inline" if you are specifying the code in the service instance.

To configure the Java Task service, you must specify settings for the following fields in Sterling Integrator:

| Field | Description |
|---|---|
| srcLocationMode | If you want to reference a file with a a filename, valid value is: File. If you want to reference specific Java source code, valid value is: Inline. |
| relativePath | If File is selected in the srcLocationMode field you must designate if there is a relativePath. If Yes, then the path is relative to the root of the Sterling Integrator installation directory. If No, the path is absolute. |
| javaFile | If File is selected in the srcLocationMode field, you must specify the file name with either a relative or absolute path. |
| javaSrc | If Inline is selected in the srcLocationMode field, you must specify the specific Java source code you want to execute.<br>**Note:** The total inline source code specified cannot be larger than 2048 characters (bytes). |

## Business Process Example

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

```
<process
name="JavaTaskDocTestFile">
<sequence>
<operation name="JavaTask">
<participant name="JavaTaskDocTestFile"/>
<output message="Xout" >
<assign to="srcLocationMode" from="'file'" />
<assign to="javaFile"
from="'/usr/local/source/DocumentCreator.java'" />
<assign to="." from="*"></assign>
</output>
<input message="Xin" >
<assign to="." from="*"></assign>
</input>
</operation>
<operation name="JavaTaskFS">
<participant name="JavaTaskFS"/>
<output message="Xout" >
<assign to="Action">FS_EXTRACT</assign>
<assign to="." from="*"></assign>
</output>
<input message="Xin" >
<assign to="." from="*"></assign>
```

```
    </input>
  </operation>
</sequence>
</process>
```

# Chapter 117. Java Database Connectivity (JDBC) Adapter

The Java Database Connectivity (JDBC) adapter enables the Translation service to communicate with JDBC-compliant databases. The adapter enables you to update or retrieve data from a JDBC-compliant database as part of a business process within the application. The following table provides an overview of the JDBC adapter:

| System name | JDBCAdapterType |
| --- | --- |
| Graphical Process Modeler (GPM) category | All Services |
| Description | Queries data from a remote database. |
| Business usage | Used to query or update data from a remote JDBC database by calling the Translator service. |
| Usage example | To perform any type of database query and return the results. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported application platforms |
| Related services | No |
| Application requirements | To use the JDBC adapter, the jdbc_customer.properties file must be edited to reference a valid data source and database. Also, the map that is going to be used by the translator must be checked in to the application. |
| Initiates business processes? | If configured to start a new business process, the results returned from the database query will be used to start a new business process. |
| Invocation | Runs as part of a business process. |
| Business process context considerations | No |
| Returned status values | Returned status values:<br>• Success – JDBC adapter finished successfully.<br>• Warning – JDBC adapter finished but with warnings.<br>• Error – JDBC adapter finished, but with errors. |
| Restrictions | None |

## How the JDBC Adapter Works

The queries you define in your map determine the data that to be retrieved or updated. You can submit any query written in SQL, stored procedures, or stored functions to a database. The business process you create then determines how the data is used.

The following figure shows how the JDBC adapter communicates with an SQL database within a business process:
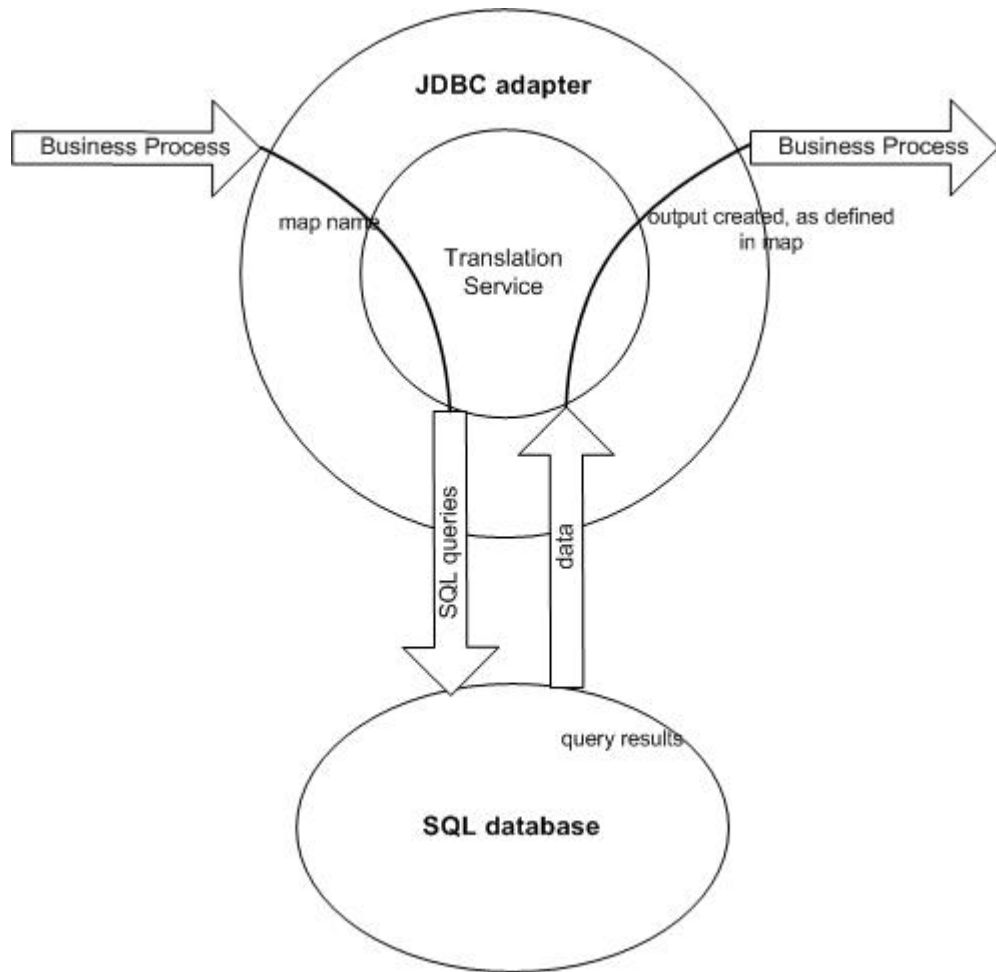
### JDBC Adapter Business Process Usage

The JDBC adapter can start a business process, or it can be used in the middle or at the end of a business process.

The following steps summarize how the JDBC adapter might be used in a business process:

1. The JDBC adapter receives a map name from the business process.
2. The adapter starts the Translation service and passes the name of the map to the translator.
3. Using the map, the translator creates the SQL commands and sends them to an SQL database.
4. The translator receives the results from an SQL database query, creates the output as defined in the map, and passes those results back to the JDBC adapter.
5. The business process advances to the next step.

### Example

For example, you have customer information stored in a confidential database. Your Sales department does not have access to the database. You can use the JDBC

adapter to provide the Sales department with access to customer information in the database and then write the information to disk using the File System adapter.

The following steps summarize the JDBC adapter data flow for this example:

1. The adapter receives the name of the map from the business process.
2. The adapter starts the Translation service and passes it the map name.
3. The Translation service executes the map and generates SQL queries to submit to the database.
4. The Translation service submits the SQL queries to the database.
5. The Translation service receives an SQL response from the database and performs another translation to establish that the response is in a format that the adapter can process.
6. The Translation service passes the translated response to the adapter.
7. The adapter sends the customer information to the next step in the business process, the File System adapter.
8. The File System adapter writes the retrieved customer information to disk for the Sales department.
9. The application performs the next activity in the business process.

## Implementing the JDBC Adapter

To implement the JDBC adapter, complete the following tasks:

1. Create a JDBC adapter configuration. For information, see *Managing Services and Adapters*.
2. Configure the JDBC adapter. For information, see *Configuring the JDBC Adapter*.
3. Install the appropriate database drivers, if necessary, on the same computer where the Map Editor was installed.
4. Create an ODBC data source. For more information about ODBC, access www.msdn.microsoft.com and locate the ODBC Programming Reference documentation.
5. Set up a connection to an external database. For information, see *Setting Up a Connection to an External Database*.
6. Create either an input or output map using the Map Editor.
7. Check in the map for versioning control.
8. Use the JDBC adapter in a business process.

## Configuring the JDBC Adapter

## Application Configuration

The following table describes the fields used to configure the JDBC adapter in your application:

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

| Field | Description |
|---|---|
| Select a Group | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |
| Start a new business process (StartNewWorkFlow) | Whether to start a new business process. Valid values are:<br><br>• Yes – This JDBC adapter starts a new business process.<br>• No – This JDBC adapter does not start a new business process. |
| Business Process | Select the business process that this adapter will start. Only applicable when **Start a new business process** is set to Yes. |
| • EDI Output Tag Delimiter (edi_output_tag_delimiter)<br>• EDI Output Segment Terminator (edi_output_segment_delimiter)<br>• EDI Output Data Element Separator (edi_output_element_delimiter)<br>• EDI Output Component Element Separator edi_output_sub-element_delimiter)<br>• EDI Output Release Character (edi_output_release_character)<br>• EDI Output Repeating Element Separator (edi_output_repeating_element_delimiter)<br>• EDI Output Decimal Character (edi_output_decimal_separator) | Values to change delimiters specified in the map, if the output side of the map is in EDI format. Optional. |
| Map Name(map_name) | Map name used by the translator. |
| Output Report To Process Data (output_report_to_process_data) | Whether to output the report to process data. Valid values are:<br><br>• Yes – Output the report to process data.<br>• No – Do no output the report to process data. |
| Sender Identity ID (SenderIdentityID) | Used by map to access trading partner codelists. |
| Receiver Identity ID (ReceiverIdentityID) | Used by map to access trading partner codelists. |
| Run as User | Enter (or select from the list) the user ID to be associated with business process instances of this service. |
| Use 24 Hour Clock Display | Select to specify times for this schedule using the 24 hour clock. Leave blank to use 12 hour clock and AM and PM. |

| Field | Description |
|---|---|
| Schedule | Information about scheduling the JDBC configuration to run and to start the specified business process. Valid values:<br><br>• Do not use schedule<br><br>  If this field is selected, this service does not start a business process and does not run on a schedule.<br><br>• Run based on timer<br><br>  Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup.<br><br>• Run daily<br><br>  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.<br><br>• Run based on day(s) of the week<br><br>  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.<br><br>• Run based on day(s) of the month<br><br>  Valid values are the day of the month (including the last day of the month (LDOM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup.<br><br>**Note:** The Schedule field only displays as an option if you set the Start a new business process parameter to "This JDBC adapter will start a new business process". |

## Parameters That Must Be Added in BPML

The following parameters can be used with the JDBC adapter by editing the BPML.

| Parameter | Description |
|---|---|
| FromSchema | Used to enable manipulation of a database schema prefix within the SQL Table/View or SQL Statement of a map. This parameter is required when overriding schema names within one or more SQL Statement fields. If the FromSchema and ToSchema parameters are not supplied, then no schema name substitution is performed.<br>**Note:** The schema search/replace is case-sensitive. |

| Parameter | Description |
|---|---|
| ToSchema | Used to enable manipulation of a database schema prefix within the SQL Table/View or SQL Statement of a map. **Note:** The schema search/replace is case-sensitive. If the FromSchema and ToSchema parameters are not supplied, then no schema name substitution is performed. If the ToSchema parameter is supplied and contains a non-empty value, then any matching schema names are changed at translation time to use the supplied ToSchema schema value as follows:<br><br>• For a SQL Statement, only schema names that match the FromSchema value will be substituted. The FromSchema parameter is required—otherwise, no schema values are substituted. To match and substitute more than one value pair, the FromSchema and ToSchema parameter strings can be delimited with an @ sign. For example:<br><br>`FromSchema="from1@from2"`<br>`ToSchema="to1@to2"`<br><br>In this example, any schema names matching "from1" are changed to "to1," and any schema names matching "from2" are changed to "to2."<br><br>For convenience, you can supply fewer ToSchema fragments than FromSchema fragments, and when there is no corresponding ToSchema fragment, the last fragment in the ToSchema string is used. For example:<br><br>`FromSchema="from1@from2@from3"ToSchema="to"`<br><br>In this example, any schema names matching "from1," "from2," or "from3" will be changed to "to."<br><br>• For a SQL Table/View, the FromSchema parameter is optional. If it is not supplied, all schema names are changed to the supplied ToSchema value. If it is supplied, the substitution occurs in the same way as it does for a SQL Statement. If the translator property **sql.driver.useIdentifierQuoteString** is set to True within customer_overrides.properties, then matching and substitution occurs with quoted schema names.<br><br>• If the ToSchema parameter is supplied but is empty (equal to "" (two double quotation marks) or '' (two single quotation marks)), then any matching schema names contained in the map are removed at translation time. |

## Setting Up a Connection to an External Database

You must set up a connection to an external database for the JDBC adapter. You can use any of the databases supported by the application for internal use (see the System Requirements documentation), or other JDBC-compliant databases, such as Sybase.

## Adding New Database Pools

To define a new database pool for use by the JDBC adapter, you must add settings for the pool to the jdbc_customer.properties.in file, which is located in the application's */install_dir*/properties directory.

In jdbc_customer.properties.in, specify the database server name, port number, database/catalog name, user ID and password. To encrypt your database password, use the encrypt_string.sh or encrypt_string.cmd utility in the bin directory. Then place the encrypted password, prefixed by an encryption indicator, in your properties file.

**CAUTION:**
**There are two jdbc_customer.properties files: jdbc_customer.properties.in, which is the "template" properties file; and jdbc_customer.properties, which is the "packaged" properties file. It is extremely important to ensure that you add the records to the template file, jdbc_customer.properties.in, not to the packaged file. Each time you run the setupfiles command in the application, all the packaged files are updated with the information contained in their template (.in) files. This means that if you make changes to the packaged file, jdbc_customer.properties, they are lost each time setupfiles runs. Always make changes to the template file, jdbc_customer.properties.in, and your changes will be maintained.**

If the database you want to connect to resides on a database server type that is not the same as the application database server type, you must also install a JDBC driver using the install3rdparty.sh or install3rdparty.cmd utility.

Select a table and column in your database to use in the test on reserve function. This function causes the application to test the database connection using a quickly run query before attempting to use it. This function ensures that idle connections are revived. The column referenced in the query should be of the type varchar and should be at least five characters in length.

## Connecting to an External Database

To connect to an external database:

1. Add the necessary records to the jdbc_customer.properties.in file found in the /install_dir/properties directory.

   **Note:** If invalid data (like ABC or 13.45) is entered in a pool setting, the setting uses its default value.

   See the examples that follow this procedure for *Oracle 8i/9i, DB2, MS SQL 2000,* and *Sybase.*

   The following table contains the parameters needed to add a new database pool to the jdbc_customer.properties.in file:

| Parameter | Description |
|---|---|
| *databasePool*.driver | JDBC driver class file for the database application. |
| *databasePool*.url | Database location (full URL as defined by the Java JDBC standards).<br>**Note:** For Oracle systems, the last segment in the URL is the Oracle SID (not the System Reference or Tnsnames entry).<br>**Note:** You can locate the Java JDBC standards on the http://www.java.sun.com Web site. |
| *databasePool*.user | Username for logging into the database. |
| *databasePool*.password | Password for logging into the database. |
| *databasePool*.maxconn | Maximum number of database connections for the connection pool. |

| Parameter | Description |
|---|---|
| *databasePool*.storedProcClassName | Specifies the class that handles stored procedure calls for the JDBC adapter. The following classes are used for the database types:<br><br>• MSSQL, Sybase, and DB2 –<br><br>com.sterlingcommerce.woodstock.util.frame.jdbc. GenericStoredProcQuery<br><br>• Oracle 8i/9i –<br><br>com.sterlingcommerce.woodstock.util.frame.jdbc. OracleNoAppStoredProcQuery<br><br>**Note:** The JDBC adapter does not support stored procedures for DB2/ iSeries, DB2/zOS, and MySQL. |
| *databasePool*.varDataClassName | Each database that the application supports handles binary objects differently. This parameter specifies the class used to handle binary data for the database. Enter the correct class for your database:<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. DB2ISeriesVarData<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. DB2VarData<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. DB2ZOSVarData<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. JConnectVarData<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. MSSQLVarData<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. MySQLVarData<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. OracleBlobVarData<br><br>• com.sterlingcommerce.woodstock.util.frame.jdbc. OracleVarData |
| *databasePool*.catalog | Database name (usually the same as the last segment of the URL) |
| *databasePool*.type | Valid values:<br>• local<br>• remote (default) |
| *databasePool*.testOnReserve | Whether to test the connection. Valid values:<br>• true<br>• false<br><br>**Note:** This function causes the application to test the database connection before attempting to use it and revives idle connections. |

| Parameter | Description |
|---|---|
| *databasePool*.testOnReserveQuery | SQL query to use when testing the connection. Select a table and column in your database to use in the test on reserve function. The column referenced in the query should be of the type varchar and should be at least five characters in length. This query needs to be executable by the *databasePool*.username account and must be a valid SQL query.<br><br>For example: `SELECT table_name FROM user_tables WHERE table_name=?`<br><br>Where ? must accept a string value. The query does not have to return a value to operate. If the query fails, the Database Pool is not activated. |
| *databasePool.testOnReserveInterval* | The minimum number of milliseconds between running testOnReserve on the same connection. The default value is 60000. Valid values:<br>• 0 – No interval and current interval is used.<br>• <= 0 – No interval.<br>• > 0 – The minimum number of milliseconds between running testOnReserve on the same connection. |
| *databasePool*.max8177RetryCount | Only used for an Oracle database, this tells the software how many times to retry if it receives an ORA-8177 error in certain situations. |
| *databasePool*.dbvendor | Enter the database name: sybase, oracle, mysql, mssql, db2, db2zos, db2iseries, or other vendor name. |
| *databasePool*.maxsize | Maximum size of the database pool. This property was previously contained in the poolManager.properties file. This value must not exceed the value specified for the databasePool.maxconn parameter in the jdbc.properties file. |
| *databasePool*.initsize | Initial size of the database pool. This property was previously contained in the poolManager.properties file. |
| *databasePool*.factory | Always enter the following:<br><br>com.sterlingcommerce.woodstock.<br>util.frame.jdbc.ConnectionFactory |

| Parameter | Description |
|---|---|
| *databasePool*.behaviour | Behavior a connection pool exhibits when it runs out of connections. This property replaces the databasePool.onEmpty property in the former poolManager.properties file. Valid values: |
| | • 0 – The pool simply returns indicating to the software to abort its current action and try again later. This value corresponds to the value return in the databasePool.onEmpty property. |
| | • 1 – The pool waits the number of milliseconds specified in databasePool.waittime for a connection to be returned before indicating to the software to abort and try again. This value corresponds to the value wait in the databasePool.onEmpty property. |
| | • 2 – The pool creates a buffered connection (a connection above the size specified in databasePool.maxsize). When using a setting of 2, the maximum number of connections for the pool is the value specified for databasePool.maxsize plus the value specified for databasePool.buffersize connections. This allows connections to be created under heavy demand. This value corresponds to the value new in the databasePool.onEmpty property. |
| *databasePool.lifespan* | The number of milliseconds a connection will live in a given pool before it needs to be removed. |
| | • 0 – (Default) No timeout. |
| | • <= 0 – No timeout. |
| | • > 0 – Number of milliseconds that a connection stay in pool. |
| *databasePool.idletimeout* | The number of milliseconds a connection can stay idle in a given pool before it needs to be removed. The default value is 86400000. Valid values: |
| | • 0 – No timeout. |
| | • <= 0 – No timeout. |
| | • > 0 – Number of milliseconds that a connection stay in pool. |
| *databasePool.housekeepinginterval* | The minimum number of milliseconds between running the housekeeping task to clean out idle connections. Valid values are any positive number. The default value is 3600000 milliseconds (1 hour). Any number less than 3600000 will cause the default of 3600000 milliseconds to be used. |
| *databasePool*.buffersize | Number of extra connections that the connection pool can create above the value specified for databasePool.maxsize to improve handling of unanticipated loads on the system. This property is only used if databasePool.behavior is set to 2. |
| *databasePool*.waittime | Amount of time (in milliseconds) to wait for a connection to become available before indicating to the software to abort the current action and try again later. This property is only used if databasePool.behavior is set to 1. |

2. Run the setupfiles.sh (UNIX) or setupfiles.cmd (Windows) utility located in the /install_dir/bin directory of the application installation directory. This updates

the "packaged" properties file, jdbc_customer.properties, with the changes from the "template" properties file, jdbc_customer.properties.in.

3. If the vendor for the connection database is not the same vendor as the database vendor used for the application database, install the appropriate JDBC driver to access the database server. Use the install3rdParty.sh (UNIX) or install3rdparty.cmd (Windows) utility located in the /install_dir/bin directory of the application's installation directory to add the JDBC driver jar file(s). Type `install3rdParty` on the command line to get a description of the parameters you can specify.

   The following examples are for a UNIX environment. The vendor name and version are the first two parameters, along with the location of the zip file containing the JDBC driver files.

   - For Oracle 9i, install the driver using the following command:

     `./install3rdParty.sh Oracle 9_2_0_5 -d` */usr/local/directory/*`oracle/9_2_0_5/classes12.zip`

   - For DB2, install the driver using the following command:

     `./install3rdParty.sh db2java 7_2 -d` */usr/local/directory/*`db2java.zip`

   - To install the jConnect driver for Sybase, refer to *Installing a Sybase Driver*.

4. Stop and restart the application to use the changed files.

## Installing a Sybase Driver

Install the jConnect driver for Sybase using the following procedure:

1. Download jConnect-5_5.zip from the Sybase web site.
2. Run the following command:

   `./install3rdParty.sh jconnect 5_5 -d` */usr/local/directory/*`jconnect/5_5/jConnect-5_5.zip`

   - If this command succeeds, you are finished with this procedure.
   - If the application reports in the system log that the driver could not be registered because the driver class cannot be found, continue with the procedure. Use the following steps to remove existing references to jConnect.

3. Stop the application.
4. Change your directory to *install_dir*/jar.
5. Delete any existing folders referencing jConnect.
6. Change your directory to *install_dir*/properties.
7. Open the files dynamicclasspath.cfg and dynamicclasspath.cfg.in. Delete any lines referencing jConnect, and save the files.
8. Create the following temporary directory:

   *install_dir*/bin/jconnect

9. Extract only the jar files from jConnect-5_5.zip to this directory.
10. Run the following command:

    `install3rdparty jconnect 5_5 -d` *install_dir*`/bin/jconnect/*.jar`

11. Check *install_dir*/jar/jconnect/5_5/*your_platform* to make sure that six jar files have been copied successfully.
12. Open the dynamicclasspath.cfg file in *install_dir*/properties and check that the following entries are included:

    ```
    VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jconn2.jar
    VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jconn2d.jar
    VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jisql.jar
    ```

```
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2d.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/ribo.jar
```

13. Open the customer.jdbc.properties.in file in *install_dir*/properties and check that the following entries are included:

```
jconnectPool.driver=com.sybase.jdbc2.jdbc.SybDriver
jconnectPool.url=jdbc:sybase:Tds:your
Hostname:4100/your DB
jconnectPool.user=your user name
jconnectPool.password=your password
jconnectPool.catalog=your  database
jconnectPool.type=local
jconnectPool.testOnReserve=true
jconnectPool.testOnReserveQuery=your Test On Reserve Query
jconnectPool.dbvendor=Sybase
jconnectPool.buffersize=500
jconnectPool.maxsize=28
jconnectPool.initsize=1
jconnectPool.behaviour=2
jconnectPool.waittime=1000
jconnectPool.storedProcClassName= com.sterlingcommerce.woodstock.
        util.frame.jdbc.SybaseStoredProcQuery
jconnectPool.varDataClassName=com.sterlingcommerce.woodstock.util.
        frame.jdbc.JConnectVarData
jconnectPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
         ConnectionFactory
```

14. Save the customer.jdbc.properties.in file and run the following command:

```
install_dir/bin/setupfiles.sh
```

This procedure should result in a successful connection to your Sybase database. However, if the database has been configured as character set ROMAN8, it is likely that you will see the following message in the application system log, because of a limitation in the Sybase driver:

```
java.sql.SQLWarning: JZ0IB: The server's
default charset of roman8 does not map to an encoding that is available
in the client Java environment. Because jConnect will not be able
to do client-side conversion, the connection is unusable and is
being closed. Try using a later Java version or try including your
Java installation's i18n.jar or charsets.jar file in the classpath
```

One resolution of this issue is to configure the primary Adaptive Server with a default character set that maps to one of the character sets supported by jConnect for JDBC (for example, UTF-8). For more details, refer to the Sybase documentation.

Another resolution of this issue is to use the open source jTDS driver from Sourceforge (sourceforge.net). To install this driver, follow these instructions:

1. Stop the application.
2. Remove references to jConnect as described previously.
3. Copy the jtds-1.2.jar file to an accessible directory on the application machine.
4. Run the following command:

   *install_dir*/bin/Install3rdparty.sh jTDS 1_2 - jar *absolutePath*/jtds-1.2.jar
5. Check that the dynamicclasspath.cfg file has picked up this change. For example, */install_dir*/jar/jTDS/1_2/*your_platform*/jtds-1.2.jar.
6. Edit the jdbc_customer.properties.in file. The definition of the pool should be similar to the following example:

```
jTDSPool.driver=net.sourceforge.jtds.jdbc.Driver
jTDSPool.url=jdbc:jtds:sybase://your Hostname:4100/answer
jTDSPool.user=your user name
jTDSPool.password=your password
jTDSPool.catalog=your database
jTDSPool.type=local
jTDSPool.testOnReserve=false
#jTDSPool.testOnReserveQuery=your Test On Reserve Query
jTDSPool.dbvendor=jtds
jTDSPool.buffersize=50
jTDSPool.maxsize=20
jTDSPool.initsize=5
jTDSPool.behaviour=2
jTDSPool.storedProcClassName=com.sterlingcommerce.woodstock.util.
       frame.jdbc.SybaseStoredProcQuery
jTDSPool.varDataClassName=com.sterlingcommerce.woodstock.util.
       frame.jdbc.GenericVarData
jTDSPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
       ConnectionFactory
```

7. Restart the application.

## Properties File Examples for Specific Databases

### Oracle 8i/9i

For Oracle 8i/9i, enter the following parameters in the jdbc_customer.properties.in file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=oracle.jdbc.OracleDriver
databasePool.url=jdbc:oracle:thin:@servername:0000:servername
databasePool.user=username
databasePool.password=password
databasePool.catalog=catalogname
databasePool.type=local
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from
      Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.max8177RetryCount=n
databasePool.dbvendor=oracle
databasePool.buffersize=n
databasePool.maxsize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=n
databasePool.storedProcClassName= com.sterlingcommerce.woodstock.util.
      frame.jdbc.OracleNoAppStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.
      frame.jdbc.OracleVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
      ConnectionFactory
```

### DB2

For DB2, enter the following parameters in the jdbc_customer.properties.in file, where databasePool is the name of the pool you are adding. Sample values are

italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

**Note:** The JDBC adapter does not support stored procedures for DB2/iSeries and DB2/zOS.

```
databasePool.driver=com.ibm.as400.access.AS400JDBCDriver
databasePool.url=jdbc:as400://serverName/DB2Database;translate
      binary=true;transaction isolation=none;
databasePool.catalog=DB2Database
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.
      frame.jdbc.DB2ISeriesVarData
databasePool.user=username
databasePool.password=password
databasePool.maxconn=20
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT SI_VERSION from SI_VERSION
      where SI_VERSION = ?
databasePool.testOnReserveInterval=60000
databasePool.blobPageSize=1024000
databasePool.compressBlob=true
databasePool.cacheps=true
databasePool.dbvendor=db2iSeries
databasePool.buffersize=500
databasePool.maxsize=20
databasePool.initsize=0
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.
      jdbc.ConnectionFactory
databasePool.behaviour=2
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=1000
```

## MS SQL 2000

For MS SQL 2000, enter the following parameters in the jdbc_customer.properties.in file, where databasePool is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
databasePool.url=jdbc:microsoft:sqlserver://servername:0000;
      DatabaseName=SQLdatabase;SelectMethod=cursor
databasePool.user=username
databasePool.password=password
databasePool.catalog=catalogname
databasePool.type=local
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from
      Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.dbvendor=mssql
databasePool.buffersize=n
databasePool.maxsize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=n
databasePool.storedProcClassName=com.sterlingcommerce.woodstock.util.
      frame.jdbc.GenericStoredProcQuery
```

```
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.
     frame.jdbc.MSSQLVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
     ConnectionFactory
```

## Sybase

For Sybase, enter the following parameters in the jdbc_customer.properties.in file, where databasePool is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=com.sybase.jdbc2.jdbc.SybDriver
databasePool.url=jdbc:sybase:Tds:servername:0000/SybaseDB
databasePool.user=username
databasePool.password=password
databasePool.catalog=catalogname
databasePool.type=local
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from
     Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.dbvendor=Sybase
databasePool.buffersize=n
databasePool.maxsize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=n
databasePool.storedProcClassName=com.sterlingcommerce.woodstock.util.
      frame.jdbc.SybaseStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.
     frame.jdbc.JConnectVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.
     jdbc.ConnectionFactory
```

## Encrypting Your Database Password

To encrypt your database password:
1. Use encrypt_string.sh (in Windows, encrypt_string.cmd).
2. When prompted, enter your external database password.
   The script returns the encrypted value for your password.
3. Place the encrypted password in your jdbc.properties.in file entry (see step 2 in the previous procedure), prefixing the encrypted password with ENCRYPTED.
   For example, myDSN.password=ENCRYPTED:rO0ABXQABkRFU2VkZXVy.

# Chapter 118. JCA Listener Service and GIS Resource Adapter

The JCA Listener service and GIS Resource adapter work together to enable you to integrate Sterling Integrator with your legacy systems. The following table provides an overview of the JCA Listener service:

| System name | SI_JCA_ADAPTER |
|---|---|
| Graphical Process Modeler (GPM) category | None |
| Description | Listens for incoming requests from Java Connector Architecture™ (JCA) requests. Runs business processes and returns resultant documents. A JCA Resource adapter (for Sterling Integrator, called the GIS Resource adapter) is a standard mechanism in J2EE™ for supplying connectivity to remote enterprise integration systems from an application server. The JCA Listener service receives information from the GIS Resource adapter and starts a business process. |
| Business usage | The GIS Resource adapter is deployed in two parts: one in a remote application server, the other in Sterling Integrator. This enables you to use an Enterprise Java Bean (EJB) that you write to start Sterling Integrator business processes directly from your application server. You can also pass documents to Sterling Integrator for processing and retrieve documents from Sterling Integrator. This enables you to integrate Sterling Integrator into your existing systems. |
| Usage example | A legacy system running on an application server (for example, JBoss™ or WebLogic®) contains data such as a purchase order or bill of materials. The legacy system can run a business process in Sterling Integrator and pass documents into Sterling Integrator for processing. The legacy system can also retrieve the results of the processing from Sterling Integrator. |
| Preconfigured? | No |
| Requires third party files? | Application server must have j2ee.jar (v1.3.1) and xerces.jar in its classpath. |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | The JCA Listener service is designed to work with the GIS Resource adapter for Sterling Integrator, which is deployed in an application server. |
| Application requirements | The Resource adapter (v 1.0) for Sterling Integrator must be deployed in an application server before using the JCA Listener service. |
| Initiates business processes? | This adapter starts business processes requested through the JCA Listener service on the remote application server. You integrate the remote application with an EJB written to access Sterling Integrator through the JCA Listener service. |
| Invocation | Does not run by a business process. |
| Business process context considerations | Any text nodes placed under the message_to_child node at the top of ProcessData will be returned to the calling EJB. |

| Returned status values | If this adapter starts from a business process, it will return an error. The completed business process returns the process ID to the Listener, which then allows for using the PID to query for the status, and to retrieve process data or documents. |
|---|---|
| Restrictions | There may be multiple configurations of this adapter, but each must listen on a unique port. |

## Requirements

To use the GIS Resource adapter, you should have knowledge of:
- JCA specification
- Enterprise Java Beans
- XML concepts
- How process data and documents are handled in Sterling Integrator

For the GIS Resource adapter to work correctly, verify that:
- You have deployed the GIS Resource adapter in an external application server.
- You have installed and configured the JCA Listener service in Sterling Integrator.
- The files j2ee.jar (version 1.3.1 or higher) and xerces.jar are in the application server classpath.

## How the GIS Resource Adapter Works

The GIS Resource adapter is deployed in a remote application server. This enables you to use a custom-written Enterprise Java Bean (EJB) to run a business process inside Sterling Integrator. You can also pass documents to Sterling Integrator for processing and retrieve documents from Sterling Integrator. This enables you to integrate Sterling Integrator into your existing systems.

The GIS Resource adapter and the JCA Listener service can be configured to run in synchronous or asynchronous mode.

The following steps summarize how the GIS Resource adapter and the JCA Listener service work in synchronous mode. In this mode, you can run a predefined Sterling Integrator business process and receive the resulting document when the business process completes.

1. A legacy system running on an external application server has a document that requires processing by Sterling Integrator.
2. The legacy system uses an EJB you write to specify the Sterling Integrator business process to run and to pass the document to the GIS Resource adapter.
3. The GIS Resource adapter transfers the document and the name of the business process to run to the JCA Listener service running in Sterling Integrator.
4. The JCA Listener service starts the specified business process and passes it the input document.
5. When the business process is completed, the resultant document is returned to the GIS Resource adapter and back to the EJB.

Using the asynchronous mode of communication, you can run a predefined Sterling Integrator business process without waiting for the business process to complete. In this case, you can go back at some later time to retrieve resulting documents and status.

**Note:** The JCA Listener service cannot run as part of a business process.

The following figure shows how the GIS Resource adapter, running on an external application server, interacts with the JCA Listener service running on Sterling Integrator.
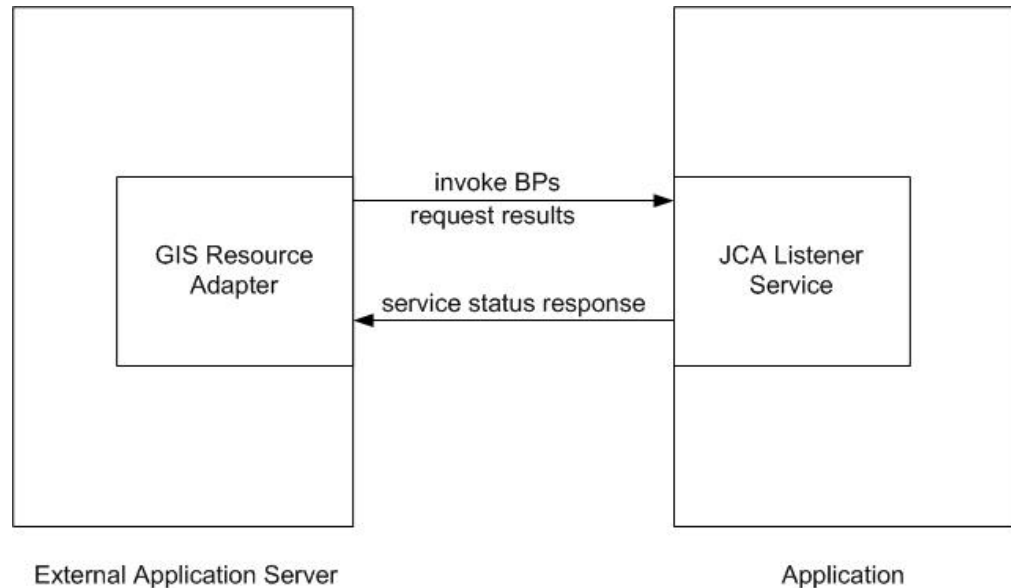


### Limitations to Full JCA Specification Support

The GIS Resource adapter does not support the following portions of the JCA specification:

- Non-managed two-tier application scenarios, that is, applications that are not managed by an application server
- Transactions
- Connection pooling, in the traditional sense, as the adapter does not persist any live connections

### Sample Business Scenario

This section describes a sample business scenario involving the JCA adapter.

You have a legacy billing system, and need to pass invoices to Sterling Integrator for processing, then receive back the results of that process.

1. Write the necessary business process in Sterling Integrator.
2. Configure the GIS Resource adapter to work with your legacy billing system server.
3. Write an EJB that passes the file from the legacy billing system to the GIS Resource adapter, and specifies the business process to run in Sterling Integrator.
4. Identify a free port on the server for the JCA Listener service to listen on.
5. Create a JCA Listener service configuration. For information, see *Managing Services and Adapters*.
6. Configure the JCA Listener service. For information, see *Configuring the JCA Listener Service*.

7. Enable the JCA Listener service and GIS Resource adapter.

8. When your billing system produces a file to be sent to Sterling Integrator, the EJB picks up the file from the legacy system and passes it and the business process name to the GIS Resource adapter.

9. The GIS Resource adapter passes the file and business process name to the JCA Listener service.

10. The business process starts, runs successfully, and puts the results in the business process context.

11. The response is passed back from the JCA Listener service to the GIS Resource adapter, then to the legacy system.

## Implementing the GIS Resource Adapter

To implement the GIS Resource adapter, complete the following tasks:

1. Deploy the GIS Resource adapter on your application server.

2. Configure the GIS Resource adapter using your application server administrator console or by editing the application server deployment descriptors.

3. Make classpath changes to support compiling external J2EE components.

4. Create a JCA Listener service configuration. For information, see *Managing Services and Adapters*.

5. Configure the JCA Listener service. For information, see *Configuring the JCA Listener Service*.

   You can configure multiple instances of this adapter, each one on a unique port.

6. Use the JCA Listener service in a business process.

## Installed Location

The resource adapter (RAR) file containing the resource adapter implementation and the deployment descriptors is available under the client/jca subdirectory. The RAR files for deploying in JBoss and in WebLogic application servers are under the JBoss and WebLogic subdirectories, respectively.

## JBoss Deployment

To deploy the GIS Resource adapter on a JBoss application server, copy the following files to the application server deployment directory:

- sijca-service.xml – JBoss deployment descriptor for the resource adapter.
- sijca_<version>_jboss.rar – Contains the resource adapter implementation classes and the standard deployment descriptor, ra.xml.

The Resource adapter can then be accessed through the JBoss JMX console.

## WebLogic Deployment

To deploy the GIS Resource adapter in a WebLogic application server, the resource adapter (RAR file) must be packaged in an Enterprise Archive (EAR) file along with the J2EE component that will access it. It must be deployed as an EAR application to establish that the CCI classes are loaded and accessible to the J2EE component.

The sijca_<version>_wl.rar contains:

- ra.xml – The standard deployment descriptor for the Resource adapter
- weblogic-ra.xml – The WebLogic deployment descriptor for the Resource adapter
- Resource adapter implementation classes

Alternatively, si_jca_<version>_client.jar can be added to the WebLogic system classpath. In that case, deployment of new versions of the resource adapter will require an application server restart. To leverage the hot-deploy capability of the WebLogic server, it is recommended that the Resource adapter be deployed as part of an EAR application.

The Resource adapter can be accessed and configured through the WebLogic server administrator console.

## Configuration

Before you can use the GIS Resource adapter, you must configure the following parameters:
- HostName – The host machine where the targeted Sterling Integrator instance is running
- PortNumber – The port number where the JCA Listener service configuration is running

These parameters can be configured through the application server administrator consoles, or by editing the Resource adapter deployment descriptors (sijca-service.xml for JBoss, and weblogic-ra.xml for WebLogic).

The user name and password for authenticating the user must be provided through the ConnectionSpec when starting getConnection() on the ConnectionFactory.

## Classpath Changes

To compile the external J2EE components that will access the GIS Resource adapter, si_jca_<version>_client.jar (under client/jca) has to be added to the classpath. The javadoc for these CCI (Common Client Interface) classes is available under the client/jca/javadoc subdirectory.

## Configuring the JCA Listener Service

To configure the JCA Listener service, you must specify settings for the following fields in 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. |

| Field | Description |
|---|---|
| Select a Group | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** For more information about groups, see *Managing Services and Adapters*. |
| Listen Port(listenPort) | Port number that this instance of the adapter will listen on. This port must be a free port. Numeric. Valid values are 1025 - 65535. Required. |

## Functions Supported for Sterling Integrator

The GIS Resource adapter enables you to use Sterling Integrator functionality, such as:

- Running a business process with or without an input document, and with or without input process data

  To run a business process in Sterling Integrator:

  1. Set the operationName in the SIInteractionSpec to InvokeBusinessProcess.
  2. To specify synchronous or asynchronous invocation, set the operationMode in the InteractionSpec as follows:

     To run the business process synchronously, set the operationMode to InteractionSpec.SYNC_SEND_RECEIVE

     To run the business process asynchronously, set the operationMode to InteractionSpec.SYNC_SEND

- Retrieving documents and output process data from completed business processes

  Any process data parameters that are included inside a message_to_child xml tag in process data of the business process that is running are returned as output process data. In the following example, param1 and param2 are returned as output process data, but not param3.

  ```
  <process_data>
  <message_to_child>
  <param1>ValueOfParam1</param1>
  <param2>ValueOfParam2</param2>
  </message_to_child>
  ...
  <param3>ValueOfParam3</param3>
  ...
  </process_data>
  ```

  To retrieve a document and/or output process data from a completed business process, set the operationName to GetBusinessProcessContext; operationMode is disregarded for this operation.

- Retrieving the state of a business process.

  To retrieve the state of a business process, set the operationName to GetBusinessProcessState; operationMode is disregarded for this operation.

## Code Example

A sample EJB illustrating the use of the GIS Resource adapter is available under the samples/jca directory.

1. Look up the Resource Adapter Connection Factory:

```
InitialContext iniCtx = new InitialContext();
/*
/ For JBoss Object ref = iniCtx.lookup("java:/SI");
*/
            // For WebLogic Object ref = iniCtx.lookup("eis/SI");
System.out.println("Found ConnectionFactory - " +
    ref.getClass().getName());
cf = (SIConnectionFactory) ref;
System.out.println("Found ConnectionFactory");
```

2. Make a connection. At this point, the GIS Resource adapter connects to the JCA Listener service on Sterling Integrator, authenticates the user, and disconnects. The user name and password come from the Sterling Integrator Accounts area:

```
SIConnectionSpec info = new SIConnectionSpec("admin","password");
Connection conn = cf.getConnection(info);
        System.out.println("Created Connection #1");
Interaction interaction = conn.createInteraction();
```

3. Create an InteractionSpec defining the operation to be run and the mode of operation.

```
InteractionSpec interSpec = new SIInteractionSpec(
                SIInteractionSpec.OP_INVOKE_BP,
            InteractionSpec.SYNC_SEND_RECEIVE);
```

4. Create a MappedRecord and set the required input data.

```
MappedRecord inRec = new SIMappedRecord();
        inRec.setRecordName("InputRecord");
inRec.setRecordShortDescription("Input record");
```

5. Set the following for passing an input document:

```
inRec.put(SIMappedRecordKeys.DOC_NAME, "DocName");
inRec.put(SIMappedRecordKeys.DOC_SUBJECT, "DocSubject");
inRec.put(SIMappedRecordKeys.DOC_TYPE, "text");
inRec.put(SIMappedRecordKeys.DOC_SUBTYPE, "plain");
        inRec.put(SIMappedRecordKeys.DOC_ENCODING, "UTF-8");
        String docBody = new
String("This is the body of the test document");
inRec.put(SIMappedRecordKeys.DOC_BODY, docBody.getBytes());
```

6. Pass any input process data as follows:

```
Hashtable
processData = new Hashtable();
processData.put("processData1", "process data value 1");
        processData.put("processData2", "process data value 2");
inRec.put(SIMappedRecordKeys.PROCESS_DATA, processData);
```

7. Set the name of the business process to run.

```
inRec.put(SIMappedRecordKeys.INVOKE_BP_NAME, "BoomerangJCA");
```

8. Execute the operation.

```
MappedRecord outRec = (MappedRecord)interaction.execute(interSpec,inRec);
```

9. Retrieve the output process data. Only parameters that are inside a message_to_child node in process data will be returned.

```
Hashtable pd = (Hashtable) outRec.get(SIMappedRecordKeys.PROCESS_DATA);
Enumeration pdKeys = pd.keys();
        while (pdKeys.hasMoreElements()) {
System.out.println("Process Data key = " + pdKeys.nextElement());
};
```

10. Release the connection.

```
interaction.close();
conn.close();
```

# Chapter 119. JMS Queue Adapter

The following table provides an overview of the JMS Queue adapter:

| System name | JMS Queue Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | All Services and Messaging > Queuing |
| Description | Exchanges messages with remote JMS Queues. Use this adapter when you want to send messages to or receive messages from a remote JMS Queue server as part of a business process within the application. The adapter can also be configured to process messages sequentially, avoiding problems encountered when business process execution depends on data captured during processing of the previous message. |
| Preconfigured? | No |
| Requires third party files? | A 3rd party jar file may be necessary if the value specified for either the **InitJndiFactory** parameter or the **Factory** parameter refers to a class that is not already included in your application installation. For example, if your application server is JBoss but you need to communicate with an external Weblogic JMS server, you need to install the jar file that includes the weblogic.jndi.WLInitialContextFactory class. You can obtain the necessary jar file from the corresponding vendor or your trading partner. |
| Platform availability | All supported platforms for this application |
| Related services | JMS Topic adapter |
| Application requirements | No |
| Initiates business processes? | Initiates a business process when configured for async receive. |
| Invocation | This adapter can only be used in a business process when configured for sending or sync receive. |

## How the JMS Queue Adapter Works

The JMS Queue adapter is a *stateful* adapter; therefore, once the adapter is started, it establishes and maintains the connection to the configured queue. The adapter can be configured to work in one of three modes: send, sync receive, or async receive.

## Send Mode

When configured for Send mode, the adapter waits to be invoked by a business process. The adapter can either send a single workflow document in one invocation or it can send multiple workflow documents in one invocation (batch mode). Each workflow document is sent as a separate message. See *Invoking Batch Sending*.

If connection to the JMS Server is lost, JMS Queue adapter attempts to reestablish connection with the JMS Server with a retry delay of 60 seconds (60000 milliseconds) between two attempts. JMS Queue adapter attempts a maximum of

twenty times to reestablish connection with the JMS Server.

## Sync Receive

When configured for Sync Receive mode, the adapter waits to be invoked by a business process. Unlike when in Async Receive mode, messages remain on the server until this adapter is invoked to receive the data. One advantage of using Sync Receive mode is that multiple messages can be received in one invocation of the adapter (batch mode). The number of messages received in one invocation can be limited, if necessary. Each message received is placed into the current workflow as a separate document. See *Invoking Batch Receiving*.

## Async Receive

When configured for Async Receive mode, the adapter cannot be invoked by a business process. When the adapter starts and the session is established, it registers an asynchronous callback listener to receive messages in one of two ways:

- Messages are received when they become available and a new workflow is started (bootstrapped) to process each message. See *Invoking Batch Receiving*.
- Messages are processed in a single thread. See the Single Thread Execution parameter under *Configuring the JMS Queue Adapter*.

## Implementing the JMS Queue Adapter

To implement the JMS Queue adapter, complete the following tasks:
1. Activate your license for the JMS Queue adapter.
2. Set up a queue in your JMS server.
3. Create a JMS Queue adapter configuration. See *Creating a Service Configuration*.
4. Configure the JMS Queue adapter. See *Configuring the JMS Queue Adapter*.
5. Create a business process that includes the JMS Queue adapter and enable it.
6. Test the business process and the adapter.
7. Run the business process.

## Configuring the JMS Queue Adapter

To configure the JMS Queue adapter, you must specify field settings in the application.

## Application Configuration

The following table describes the fields used to configure the JMS Queue adapter.

**Note:** The field names in parentheses represent the corresponding field names in the Graphical Process Modeler. This information is provided for your reference.

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

| Field | Description |
|---|---|
| Select a Group | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** See *Using Service Groups*. |
| Connection Type | Whether or not the adapter uses JNDI lookup to connect to the remote JMS Queue server. Valid values are:<br><br>• Using Jndi – Uses JNDI lookup.<br>• Using Non-Jndi – Routes to the connection factory directly. Used to connect to JMS servers which also support non-JNDI connections for JMS, such as Sonic MQ and ActiveMQ. |
| Initial Context Factory (InitJndiFactory) | Initial context factory for connecting to the remote JMS Queue server. Used for JNDI lookup. Example: weblogic.jndi.WLInitialContextFactory. Required. |
| URL (JndiUrl) | (JNDI only) Uniform Resource Locator of the application server that listens for connection requests. Required. |
| Broker URL (BrokerURL) | (non-JNDI only) Universal Resource Locator of the application server that listens for connection requests. Required. |
| Remote Queue name (RemoteQueueTopicName) | Name of the remote JMS Queue that you want to exchange messages with. Required. |
| Remote Queue Connection Factory (Factory) | Encapsulates connection configuration information and enables JMS applications to create a connection with predefined attributes. Defines and configures one or more connection factories, and the JMS server adds them to the JNDI space during startup. The default is javax.jms.QueueConnectionFactory. Required. |
| Remote User Name (Username) | User name for accessing the JMS Server. Required if the JMS Server requires security credentials. |
| Remote Password (Password) | Password for accessing the JMS Server. Required if the JMS Server requires security credentials. |
| Connection User Name | Authentication user ID when security is enabled. |
| Connection Password | Password for the authentication user ID. |
| Turn on debug messages (Debug) | Whether to log debug messages for this adapter instance. Required. Valid values:<br><br>• Yes – Debug messages will be logged.<br>• No – Debug messages will not be logged. |

| Field | Description |
|---|---|
| Queue Type (Action) | Type of queue to access. Required. Valid values are:<br><br>• Queue Send – Send messages.<br><br>• Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).<br><br>• Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message. |
| Message Type (Payload) | Type of message to send. Used only if queue type is Queue Send. Valid values are:<br><br>• BytesMessage<br><br>• ObjectMessage<br><br>• StreamMessage<br><br>• TextMessage |
| Bootstrap Workflow (InitialWorkFlowId) | Business process to initiate when data is received. Used only if queue type is Queue Receive Async. Required. |
| Document Storage Type (docStorageType) | Defines how the document will be stored in the system. Used only if queue type is Queue Receive Async. Required. Valid values:<br><br>• System Default<br><br>• Database<br><br>• File System<br><br>**Note:** See *Selecting a Document Storage Method for Bootstrap Adapters*. |
| Single Thread Execution | Processes a message completely before beginning the processing of the subsequent message. Preferred mode for processes that require sequential processing. However, processing will be slower. Valid values: No, Yes<br>**Note:** Single thread execution sets the value for Max Bootstrap threads to one, even if the user has configured another value. |
| Maximum Bootstrap Threads (MaxThreads) | Maximum number of threads used when receiving files and starting business processes. Used only if queue type is Queue Receive Async. Each message received uses one thread. Default is 10. Optional.<br>**Note:** Single thread execution sets the value for Max Bootstrap threads to one, even if the user has configured another value |
| Buffer Size (BufferSize) | Size of the buffer when receiving data. Used only if queue type is Queue Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional. |

| Field | Description |
|---|---|
| Document Filename (OutputFileName) | If you choose Queue Receive Async as the queue type for the JMS Queue adapter, then you can specify a file name for the data that the JMS Queue receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form *<nodename>_yyyymmddhhmmsslll*. |
| Connection retry attempts (RetryCount) | Maximum number of connection retry attempts. Used only if queue type is Queue Receive Async. Specify -1 for an infinite number of retry attempts. Default is 20. Optional. |
| Delay between retries (RetrySleep) | Number of milliseconds to wait between retry attempts. Default is 300000 ms (5 minutes). Used only if queue type is Queue Receive Async. Optional. |
| Notification Workflow (NotifyWorkFlow) | Business process initiated by the JMS Queue adapter if the maximum number of connection retries specified in **Connection retry attempts** is exceeded. Used only if queue type is Queue Receive Async. Required. If the adapter does not initiate a business process, select Not Applicable. |
| User | User ID to use for running the adapter. Select a user ID from the list. Valid values: Any valid user ID for your application **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. |
| Jar Locations | Optional. Specify the preferred libraries of the jar files to be loaded with the JMS Queue adapter. You must specify the full path of the location of the jar files. Use semicolon (;) to separate multiple paths. |

## Graphical Process Modeler Configuration

For the JMS Queue adapter, there are no fields required to be configured in the GPM.

## Parameters Passed From Business Process to Service

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

| Parameter | Description |
|---|---|
| batchSndFilter | Optional. Only used when sending. If specified in the business process, triggers batch mode sending based on the documents that match the filter. You can use an asterisk '*' in the filter as a wildcard. |
| batchRcvLimit | Optional. Only used when receiving synchronously. If specified in the business process, the number of messages received is limited to the number specified. If not specified, all messages available are received. |

| Parameter | Description |
|---|---|
| batchRcvTimeout | Optional. Only used when receiving synchronously. If specified in the business process, it overrides the default receive timeout. If not specified, the default timeout is 2000 milliseconds (2 seconds). |

## Setting JMS Header Object Properties

When sending, you can set JMS object properties within the JMS header that are not part of the payload data. You can specify name/value pairs during runtime within the BPML. Because the user defined name/value pairs are unknown ahead of time, they cannot be set in the application or GPM configuration so they must be manually added directly in the BPML. The JMS Queue adapter will look in ProcessData for the XML node name JMSetProperty and use any child nodes it finds to set the name/value pairs. There is a list of reserved property names that will set specific JMS message properties. An example of the ProcessData XML tree would look like this:

```
<ProcessData>
  <JMSetProperty>
    <somename1>somevalue1</somename1>
    <somename2>somevalue2</somename2>
Reserved names that set specific JMS message properties
    <correlationID>someStringValue</correlationID >
    <deliveryMode>someIntegerValue</deliveryMode>
        <destination>someQueueName</destination>
    <expiration>someLongValue</expiration>
    <messageID>someStringValue</messageID>
    <priority>someIntegerValue</priority>
    <redelivered>someBooleanValue(true/false)</redelivered>
    <replyTo>someQueueName</replyTo>
    <timestamp>someLongValue</timestamp>
    <type>someStringValue</type>
  </JMSetProperty>
</ProcessData>
```

An example of BPML that could be used to set these ProcessData name/value pairs follows:

```
<assign to="JMSetProperty/somename1" from="'somevalue1'" append="true"/>
<assign to="JMSetProperty/somename2" from="'somevalue2'" append="true"/>
```

When receiving, the JMS Queue adapter will set ProcessData items for all the JMS header fields and any object properties. Any object properties set in the JMS header will be put into ProcessData with the node name of JMS. For example, if there is a property called *somename* with a value of *somevalue*, ProcessData will contain JMS/somename with the corresponding value:

```
<JMS>
  <somename>somevalue</somename>
</JMS>
```

In addition to the user defined properties, the JMS Queue adapter will also set the following JMS header fields in ProcessData (if they are not null):

- JMS/correlationID
- JMS/deliveryMode
- JMS/destination
- JMS/expiration
- JMS/messageID

- JMS/priority
- JMS/redelivered
- JMS/replyTo
- JMS/timestamp
- JMS/type

The JMSetProperty can be used as a global property (under the ProcessData node) or a local property (under individual documents). Local JMSetProperty parameters override any global parameters and are useful when sending in batch mode. In the below example, the global JMSetProperty has a parameter called "test" with a value of zero. Since the PrimaryDocument does not have a local JMSetProperty, it uses the global one. However, since doc1, doc2, and doc3 have local JMSetProperty parameters, they use the local parameters.

```
<ProcessData>
  <JMSetProperty>
    <test>0</test>
  </JMSetProperty>
  <PrimaryDocument SCIObjectID="1833955:1063b363ed5:-774a"/>
  <doc1 SCIObjectID="1833955:1063b363ed5:-774b">
    <JMSetProperty>
      <test>1</test>
    </JMSetProperty>
  </doc1>
  <doc2 SCIObjectID="1833955:1063b363ed5:-774c">
    <JMSetProperty>
      <test>2</test>
    </JMSetProperty>
  </doc2>
  <doc3 SCIObjectID="1833955:1063b363ed5:-774d">
    <JMSetProperty>
      <test>3</test>
    </JMSetProperty>
  </doc3>
</ProcessData>
```

## Invoking Batch Sending

If a business process contains multiple documents in ProcessData, the JMS adapter can be invoked once with the workflow parameter **batchSndFilter**, which enables the adapter to send multiple messages for each of the documents that match the batchSndFilter criteria.

To invoke batch sending:

1. You do not need to make changes to the main adapter configuration; just add the appropriate assignment to the business process in the JMS adapter invocation step.

An example ProcessData for the example BPMLs below would look like this:

```
<ProcessData>
  <PrimaryDocument SCIObjectID="fe64b9:1060cac437b:-6a2a"/>
  <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2b"/>
  <XYZ>
    <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2c"/>
    <doc2 SCIObjectID="fe64b9:1060cac437b:-6a2d"/>
    <doc3 SCIObjectID="fe64b9:1060cac437b:-6a2e"/>
  </XYZ>
</ProcessData>
```

### Example 1

Sends all documents in ProcessData (including the PrimaryDocument). In this example, all five documents in ProcessData above are sent.

```
<operation name="JMS batch send">
  <participant name="JMSadapter"/>
  <output message="toService">
    <assign to="." from="*"/>
    <assign to="batchSndFilter" from="'*'"/>
  </output>
  <input message="fromService">
    <assign to="." from="*"/>
  </input>
</operation>
```

### Example 2

Sends all documents that begin with "doc" under the XYZ node. In this example, only three documents in the ProcessData above are sent.

```
<operation name="JMS batch send">
  <participant name="JMSadapter"/>
  <output message="toService">
    <assign to="." from="*"/>
    <assign to="batchSndFilter" from="'XYZ/doc*'"/>
  </output>
  <input message="fromService">
    <assign to="." from="*"/>
  </input>
</operation>
```

### Invoking Batch Receiving

The type of receive adapter you choose to use is based on your business needs. If you are processing a large volume of messages, you may find that batching them is more efficient than bootstrapping one workflow for every message.

For the JMS Queue adapter, there are two types of receive queues:

- Queue Receive Async – Registers a listener to the queue so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message. The business process that the adapter is going to bootstrap should be in sync mode.
- Queue Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).

Additionally, there are two business process parameters associated with Sync Receive (batch receive):

- batchRcvLimit – (Optional) If used, this parameter limits the number of messaged batched into the bootstrapped workflow. Default = no limit.
- batchRcvTimeout – (Optional) If used, this parameter specifies how long the adapter waits without receiving a message before ending. Default = 2000 (milliseconds)

Once a Sync Receive adapter completes the receive process, it creates the following information in ProcessData for the current workflow that invoked the adapter:

- JMS/DocumentCount – This parameter is always created to show how many documents were created from messages received, even if zero messages were received.
- JMS/Document*xxx* – For every message received, a document is created under the JMS node and then sequentially numbered starting with one (that is, Document1, Document2, and so forth).

Another difference between Async Receive mode and Sync Receive mode is where the message metadata is stored in ProcessData. In Async Receive mode, it only creates one document (the PrimaryDocument), so all the metadata is stored as JMS/*metadataName* . However, in Sync Receive mode, the metadata is stored under each document as JMS/*documentName*/+ as shown in the example below.

Example of ProcessData after a batch receive was performed:

```
<ProcessData><JMS>
    <DocumentCount>3</DocumentCount>
    <Document1 SCIObjectID="1833955:1060de6d03d:-697b">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156360801</messageID>
      <priority>4</priority>
      <timestamp>1125515636080</timestamp>
    </Document1>
    <Document2 SCIObjectID="1833955:1060de6d03d:-6978">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361102</messageID>
      <priority>4</priority>
      <timestamp>1125515636110</timestamp>
    </Document2>
    <Document3 SCIObjectID="1833955:1060de6d03d:-6975">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361243</messageID>
      <priority>4</priority>
      <timestamp>1125515636124</timestamp>
    </Document3>
  </JMS>
</ProcessData>
```

# Chapter 120. JMS Topic Adapter

The following table provides an overview of the JMS Topic adapter:

| System name | JMS Topic Adapter |
|---|---|
| Graphical Process Modeler (GPM) category | All Services and Messaging > Queuing |
| Description | Exchanges messages with remote JMS topics. Use this adapter when you want to send messages to or receive messages from a remote JMS Topic server as part of a business process within your application. The adapter can also be configured to process messages sequentially, avoiding potential problems when business process execution depends on data captured during processing of the previous message. |
| Preconfigured? | No |
| Requires third party files? | A 3rd party jar file may be necessary if the value specified for either the **InitJndiFactory** parameter or the **Factory** parameter refers to a class that is not already included in the installation of your application. For example, if your application server is JBoss but you need to communicate with an external Weblogic JMS server, you need to install the jar file that includes the weblogic.jndi.WLInitialContextFactory class. You can obtain the necessary jar file from the corresponding vendor or your trading partner. |
| Platform availability | All supported platforms for your application |
| Related services | JMS Queue adapter |
| Application requirements | No |
| Initiates business processes? | Initiates a business process when configured for async receive. |
| Invocation | This adapter can only be used in a business process when configured for sending or sync receive. |

## How the JMS Topic Adapter Works

The JMS Topic adapter is a *stateful* adapter; therefore, once the adapter is started, it establishes and maintains the connection to the configured Topic. The adapter can be configured to work in one of three modes: send, sync receive, or async receive.

## Send Mode

When configured for Send mode, the adapter waits to be invoked by a business process. The adapter can either send a single workflow document in one invocation or it can send multiple workflow documents in one invocation (batch mode). Each workflow document is sent as a separate message. See *Invoking Batch Sending*.

If connection to the JMS Server is lost, JMS Topic adapter attempts to reestablish connection with the JMS Server with a retry delay of 60 seconds (60000 milliseconds) between two attempts. JMS Topic adapter attempts a maximum of twenty times to reestablish connection with the JMS Server.

## Sync Receive

When configured for Sync Receive mode, the adapter waits to be invoked by a business process. Unlike when in Async Receive mode, messages remain on the server until this adapter is invoked to receive the data. One advantage of using Sync Receive mode is that multiple messages can be received in one invocation of the adapter (batch mode). The number of messages received in one invocation can be limited, if necessary. Each message received is placed into the current workflow as a separate document. See *Invoking Batch Receiving*.

## Async Receive

When configured for Async Receive mode, the adapter cannot be invoked by a business process.

When the adapter starts and the session is established, it registers an asynchronous callback listener to receive messages in one of two ways:

- Messages are received when they become available and a new workflow is started (bootstrapped) to process each message. See *Invoking Batch Receiving*.
- Messages are processed in a single thread. See the Single Thread Execution parameter under *Configuring the JMS Topic Adapter*.

## Implementing the JMS Topic Adapter

To implement the JMS Topic adapter, complete the following tasks:

1. Activate your license for the JMS Topic adapter.
2. Set up a topic in your JMS server.
3. Create a JMS Topic adapter configuration. See *Creating a Service Configuration*.
4. Configure the JMS Topic adapter. See *Configuring the JMS Topic Adapter*.
5. Create a business process that includes the JMS Topic adapter and enable it.
6. Test the business process and the adapter.
7. Run the business process.

## Configuring the JMS Topic Adapter

To configure the JMS Topic adapter, you must specify field settings in your application.

## Configuration

The following table describes the fields used to configure the JMS Topic adapter:

**Note:** The field names in parentheses represent the corresponding field names in the Graphical Process Modeler. This information is provided for your reference.

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

| Field | Description |
|---|---|
| Select a Group | Select one of the options:<br><br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• 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.<br><br>**Note:** See *Using Service Groups*. |
| Connection Type | Whether or not the adapter uses JNDI lookup to connect to the remote JMS Topic server. Valid values are:<br><br>• Using Jndi – Uses JNDI lookup.<br>• Using Non-Jndi – Routes to the connection factory directly. Used to connect to JMS servers which also support non-JNDI connections for JMS, such as Sonic MQ and Active MQ. |
| Initial Context Factory (InitJndiFactory) | Initial context factory for connecting to the remote JMS Topic server. Used for JNDI lookup. Example: weblogic.jndi.WLInitialContextFactory. Required. |
| URL (JndiUrl) | (JNDI only) Uniform Resource Locator of the application server that listens for connection requests. Required. |
| Broker URL (BrokerURL) | (non-JNDI only) Universal Resource Locator of the application server that listens for connection requests. |
| Remote Topic name (RemoteQueueTopicName) | Name of the remote JMS Topic that you want to exchange messages with. Required. |
| Remote Topic Connection Factory (Factory) | Encapsulates connection configuration information and enables JMS applications to create a connection with predefined attributes. Defines and configures one or more connection factories, and the JMS server adds them to the JNDI space during startup. The default is javax.jms.TopicConnectionFactory. Required. |
| Remote User Name (Username) | User name for accessing the JMS Server. Required if the JMS Server requires security credentials. |
| Remote Password (Password) | Password for accessing the JMS Server. Required if the JMS Server requires security credentials. |
| Connection User Name | Authentication user ID when security is enabled. |
| Connection Password | Password for the authentication user ID when security is enabled. |
| Turn on debug messages (Debug) | Whether to log debug messages for this adapter instance. Required. Valid values:<br><br>• Yes – Debug messages will be logged.<br>• No – Debug messages will not be logged. |

| Field | Description |
|---|---|
| Topic Type (Action) | Type of topic to access. Valid values are:<br><br>• Topic Send – Sends messages.<br><br>• Topic Receive Sync – Must be called by a business process for the adapter to poll for any available messages. But, instead of bootstrapping one workflow per message (such as the Async Receive adapter does), the Sync Receive adapter will create a separate workflow document for each message and place them all into the current workflow (no bootstrapping occurs).<br><br>• Topic Receive Async – Registers a listener to the topic so that when messages are available they are received immediately, or pushed down to the adapter, and a new workflow is bootstrapped to handle that single message. |
| Message Type (Payload) | Type of message to send. Used only if topic type is Topic Send. Valid values are:<br><br>• BytesMessage<br><br>• ObjectMessage<br><br>• StreamMessage<br><br>• TextMessage |
| Bootstrap Workflow (InitialWorkFlowId) | Business process to initiate when data is received. Used only if topic type is Topic Receive Async. Required. |
| Document Storage Type (docStorageType) | Defines how the document will be stored in the system. Used only if topic type is Topic Receive Async. Required. Valid values:<br><br>• System Default<br><br>• Database<br><br>• File System<br><br>**Note:** See *Selecting a Document Storage Method for Bootstrap Adapters*. |
| Single Thread Execution | Processes a message completely before beginning the processing of the subsequent message. Preferred mode for processes that require sequential processing. However, processing will be slower. Valid values: Yes, No<br>**Note:** Single thread execution sets the value for Max Bootstrap Threads to one, even if the user has configured another value. |
| Maximum Bootstrap Threads (MaxThreads) | Maximum number of threads used when receiving files and starting business processes. Used only if topic type is Topic Receive Async. Each message received uses one thread. Default is 10. Optional.<br>**Note:** Single thread execution sets the value for Max Bootstrap threads to one, even if the user has configured another value. |
| Buffer Size (BufferSize) | Size of the buffer when receiving data. Used only if topic type is Topic Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional. |

| Field | Description |
|---|---|
| Document Filename (OutputFileName) | If you choose Topic Receive Async as the topic type for the JMS Topic adapter, then you can specify a file name for the data that the JMS Topic receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form *nodename_yyyymmddhhmmsslll*. |
| Connection retry attempts (RetryCount) | Maximum number of connection retry attempts. Used only if topic type is Topic Receive Async. Specify -1 for an infinite number of retry attempts. Default is 20. Optional. |
| Delay between retries (RetrySleep) | Number of milliseconds to wait between retry attempts. Default is 300000 ms (5 minutes). Used only if topic type is Topic Receive Async. Optional. |
| Notification Workflow (NotifyWorkFlow) | Business process initiated by the JMS Topic adapter if the maximum number of connection retries specified in **Connection retry attempts** is exceeded. Used only if topic type is Topic Receive Async. Required. If the adapter does not initiate a business process, select Not Applicable. |
| User | User ID to use for running the adapter. Select a user ID from the list. Valid values: Any valid application user ID **Note:** This parameter allows someone who doesn't have rights to a specific business process to run it. If you select **Admin** as the user ID, you will inherit Administrative rights (for this run of the business process only), and enable the scheduled run. |
| Jar Locations | Optional. Specify the preferred libraries of the jar files to be loaded with the JMS Topic adapter. You must specify the full path of the location of the jar files. Use semicolon (;) to separate multiple paths. |

## Parameters Passed From Business Process to Adapter

The following table contains the parameters passed from the business process to the JMS Topic adapter:

| Parameter | Description |
|---|---|
| batchSndFilter | Optional. Only used when sending. If specified in the business process, triggers batch mode sending based on the documents that match the filter. You can use an asterisk '*' in the filter as a wildcard. |
| batchRcvLimit | Optional. Only used when receiving synchronously. If specified in the business process, the number of messages received is limited to the number specified. If not specified, all messages available are received. |
| batchRcvTimeout | Optional. Only used when receiving synchronously. If specified in the business process, it overrides the default receive timeout. If not specified, the default timeout is 2000 milliseconds (2 seconds). |

## Setting JMS Header Object Properties

When sending, you can set JMS object properties within the JMS header that are not part of the payload data. You can specify name/value pairs during runtime

within the BPML. Because the user defined name/value pairs are unknown ahead of time, they cannot be set in the application or GPM configuration so they must be manually added directly in the BPML. The JMS Topic adapter will look in ProcessData for the XML node name JMSetProperty and use any child nodes it finds to set the name/value pairs. There is a list of reserved property names that will set specific JMS message properties. An example of the ProcessData XML tree would look like this:

```
<ProcessData>
  <JMSetProperty>
    <somename1>somevalue1</somename1>
    <somename2>somevalue2</somename2>
Reserved names that set specific JMS message properties
    <correlationID>someStringValue</correlationID >
    <deliveryMode>someIntegerValue</deliveryMode>
    <destination>someTopicName</destination>
    <expiration>someLongValue</expiration>
    <messageID>someStringValue</messageID>
    <priority>someIntegerValue</priority>
    <redelivered>someBooleanValue(true/false)</redelivered>
    <replyTo>someTopicName</replyTo>
    <timestamp>someLongValue</timestamp>
    <type>someStringValue</type>
  </JMSetProperty>
</ProcessData>
```

An example of BPML that could be used to set these ProcessData name/value pairs follows:

```
<assign to="JMSetProperty/somename1" from="'somevalue1'" append="true"/>
<assign to="JMSetProperty/somename2" from="'somevalue2'" append="true"/>
```

When receiving, the JMS Topic adapter will set ProcessData items for all the JMS header fields and any object properties. Any object properties set in the JMS header will be put into ProcessData with the node name of JMS. For example, if there is a property called *somename* with a value of *somevalue*, ProcessData will contain JMS/somename with the corresponding value:

```
<JMS>
  <somename>somevalue</somename>
</JMS>
```

In addition to the user defined properties, the JMS Topic adapter will also set the following JMS header fields in ProcessData (if they are not null):

- JMS/correlationID
- JMS/deliveryMode
- JMS/destination
- JMS/expiration
- JMS/messageID
- JMS/priority
- JMS/redelivered
- JMS/replyTo
- JMS/timestamp
- JMS/type

The JMSetProperty can be used as a global property (under the ProcessData node) or a local property (under individual documents). Local JMSetProperty parameters override any global parameters and are useful when sending in batch mode. In the below example, the global JMSetProperty has a parameter called "test" with a

value of zero. Since the PrimaryDocument does not have a local JMSetProperty, it uses the global one. However, since doc1, doc2, and doc3 have local JMSetProperty parameters, they use the local parameters.

```
<ProcessData>
  <JMSetProperty>
    <test>0</test>
  </JMSetProperty>
  <PrimaryDocument SCIObjectID="1833955:1063b363ed5:-774a"/>
  <doc1 SCIObjectID="1833955:1063b363ed5:-774b">
    <JMSetProperty>
      <test>1</test>
    </JMSetProperty>
  </doc1>
  <doc2 SCIObjectID="1833955:1063b363ed5:-774c">
    <JMSetProperty>
      <test>2</test>
    </JMSetProperty>
  </doc2>
  <doc3 SCIObjectID="1833955:1063b363ed5:-774d">
    <JMSetProperty>
      <test>3</test>
    </JMSetProperty>
  </doc3>
</ProcessData>
```

## Invoking Batch Sending

If a business process contains multiple documents in ProcessData, the JMS adapter can be invoked once with the workflow parameter **batchSndFilter**, which enables the adapter to send multiple messages for each of the documents that match the batchSndFilter criteria.

To invoke batch sending:

1. You do not need to make changes to the main adapter configuration; just add the appropriate assignment to the business process in the JMS adapter invocation step.

An example ProcessData for the example BPMLs below would look like this:

```
<ProcessData>
  <PrimaryDocument SCIObjectID="fe64b9:1060cac437b:-6a2a"/>
  <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2b"/>
  <XYZ>
    <doc1 SCIObjectID="fe64b9:1060cac437b:-6a2c"/>
    <doc2 SCIObjectID="fe64b9:1060cac437b:-6a2d"/>
    <doc3 SCIObjectID="fe64b9:1060cac437b:-6a2e"/>
  </XYZ>
</ProcessData>
```

## Example 1

Sends all documents in ProcessData (including the PrimaryDocument). In this example, all five documents in ProcessData above are sent.

```
<operation name="JMS batch send">
  <participant name="JMSadapter"/>
  <output message="toService">
    <assign to="." from="*"/>
    <assign to="batchSndFilter" from="'*'"/>
  </output>
```

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

### Example 2

Sends all documents that begin with "doc" under the XYZ node. In this example,
only three documents in the ProcessData above are sent.

```
<operation name="JMS batch send">
  <participant name="JMSadapter"/>
  <output message="toService">
    <assign to="." from="*"/>
    <assign to="batchSndFilter" from="'XYZ/doc*'"/>
  </output>
  <input message="fromService">
    <assign to="." from="*"/>
  </input>
</operation>
```

### Invoking Batch Receiving

The type of receive adapter you choose to use is based on your business needs. If
you are processing a large volume of messages, you may find that batching them
is more efficient than bootstrapping one workflow for every message.

For the JMS Topic adapter, there are two types of receive topics:
- Topic Receive Async – Registers a listener to the topic so that when messages are
  available they are received immediately, or pushed down to the adapter, and a
  new workflow is bootstrapped to handle that single message. The business
  process that the adapter is going to bootstrap should be in sync mode.
- Topic Receive Sync – Must be called by a business process for the adapter to poll
  for any available messages. But, instead of bootstrapping one workflow per
  message (such as the Async Receive adapter does), the Sync Receive adapter will
  create a separate workflow document for each message and place them all into
  the current workflow (no bootstrapping occurs).

Additionally, there are two business process parameters associated with Sync
Receive (batch receive):
- batchRcvLimit – (Optional) If used, this parameter limits the number of
  messaged batched into the bootstrapped workflow. Default = no limit.
- batchRcvTimeout – (Optional) If used, this parameter specifies how long the
  adapter waits without receiving a message before ending. Default = 2000
  (milliseconds)

Once a Sync Receive adapter completes the receive process, it creates the following
information in ProcessData for the current workflow that invoked the adapter:
- JMS/DocumentCount – This parameter is always created to show how many
  documents were created from messages received, even if zero messages were
  received.
- JMS/Documentxxx – For every message received, a document is created under
  the JMS node and then sequentially numbered starting with one (that is,
  Document1, Document2, and so forth).

Another difference between Async Receive mode and Sync Receive mode is where
the message metadata is stored in ProcessData. In Async Receive mode, it only

creates one document (the PrimaryDocument), so all the metadata is stored as
JMS/*metadataName*. However, in Sync Receive mode, the metadata is stored under
each document as JMS/*documentName*/*metadataName* as shown in the example
below.

Example of ProcessData after a batch receive was performed:

```
<ProcessData>
  <JMS>
    <DocumentCount>3</DocumentCount>
    <Document1 SCIObjectID="1833955:1060de6d03d:-697b">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testtopic</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156360801</messageID>
      <priority>4</priority>
      <timestamp>1125515636080</timestamp>
    </Document1>
    <Document2 SCIObjectID="1833955:1060de6d03d:-6978">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testtopic</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361102</messageID>
      <priority>4</priority>
      <timestamp>1125515636110</timestamp>
    </Document2>
    <Document3 SCIObjectID="1833955:1060de6d03d:-6975">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
        <destination>testtopic</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361243</messageID>
      <priority>4</priority>
      <timestamp>1125515636124</timestamp>
    </Document3>
  </JMS>
</ProcessData>
```

# Chapter 121. JMS 1.1 Acquire Connection and Session Service

The JMS 1.1 Acquire Connection and Session service opens a connection and session with a remote JMS server. This service is used as part of a business process with other services like JMS 1.1 Send Message, Receive Message, Request Reply, and Release Connection services. The following table provides an overview of the service:

| System Name | JMS1.1 Acquire Connection And Session service |
|---|---|
| Graphical Process Modeler (GPM) categories | All Services |
| Description | The JMS 1.1 Acquire Connection and Session service opens a connection and session with a remote JMS Server. |
| Business usage | The JMS 1.1 Acquire Connection and Session service opens a connection and session with a remote JMS Server. |
| Usage example | Use this service in a business process to acquire a connection and session which will be used by the other JMS services. |
| Preconfigured? | No |
| Requires third party files? | The JMS provider jar must be provided to the system, using one of the following methods:<br><br>• Install using the install3rdParty script<br><br>• Load using the custom classloader<br><br>For JBoss and WebLogic, you must use the custom classloader as these vendor jars contain classes that already exist in system classloader.<br><br>Before you install/load the jar file, you need to know:<br><br>• Remote JMS server context factory<br><br>• URL (IP address and port)<br><br>• Connection factory name<br><br>• User name and password (if applicable)<br><br>• Public key for the CA or trusted certificates (if used) from the trading partner for access to their JMS server<br><br>• System certificate (if used)<br><br>For example, if you need to communicate to an external WebLogic JMS server, you need the jar file that includes the weblogic.jndi.WLInitialContextFactory class. You would obtain the jar file from the corresponding vendor or your trading partner. |
| Platform availability | All supported platforms |

| Related services | This service must be used as the first JMS 1.1 service in a business process to open the session. Related services include: |
|---|---|
| | • JMS1.1 Receive Message Service – used to get messages synchronously from a queue or topic. |
| | • JMS1.1 Send Message Service - used to send messages. |
| | • JMS1.1 Release Connection And Session Service - used to release the connection and session. |
| Application requirements | None |
| Initiates business processes? | No |
| Invocation | This service is invoked as part of a business process. |
| Business process context considerations | The service can either be configured from the UI or through the GPM. The connection and session related parameters are set in this service and are used to send, receive, request/reply and release services. |
| | If the Acquire Connection parameters are set in the GPM, they take precedence over the UI parameters. |
| Returned status values | None |
| Restrictions | None |
| Persistence level | Minimum |
| Testing considerations | Enable JMS logging from the Admin Console by selecting Operations > System > Logs. |

## Configuring the JMS 1.1 Acquire Connection and Session Service

You will need to create a configuration of the JMS 1.1 Acquire Connection and Session service in the Admin Console. You must specify field settings for the service. You can set parameters at the service configuration level in the Admin Console and at the business process level in the GPM. The GPM parameter values override any parameters set in the Admin Console.

Use the following field definitions to create a new configuration of the JMS 1.1 Acquire Connection and Session service.

| UI Field Name | GPM Field Name | Description |
|---|---|---|
| Name | | Unique name for the service configuration. Required. |
| Description | | Description for the service configuration, for reference purposes. Required. |

| UI Field Name | GPM Field Name | Description |
|---|---|---|
| Select a Group | | Select one of the options:<br>• None – Do not include the configuration in a service group at this time.<br>• 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.)<br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |
| | Config | Select the name of the service configuration from the list. |
| Connection Type | CONN_TYPE | Defines whether to use JNDI or not. Required. Valid values are:<br>• Using JNDI (Default) - Use JNDI when the JMS administered objects (such as, Connection Factory and Destination) needs to be looked up from the JNDI of the JMS provider. This can be useful if the provider wants to control the creation of these administered objects rather than JMS clients and wants to share the same across all the clients. Some vendors (WebLogic and JBoss) only support JNDI mode.<br>• Using Non-JNDI - Use when the client wants to create the administered objects in the client code. Several JMS vendors do not support Non-JNDI.<br>If you need additional information to determine which connection type is supported by your vendor, consult the vendor documentation. |
| Pool Name | POOL_NAME | Name of the pool. Required. |
| Context Factory | CONTEXT_FACTORY | Context Factory name as provided by the JMS Provider. Required. Only displayed if using JNDI. |
| URL | BROKER_URL | URL needed to connect with the JMS provider. Required. Only displayed if using JNDI. |
| Broker URL | BROKER_URL | URL needed to connect with the JMS provider. Required. Only displayed if using Non-JNDI. |

| UI Field Name | GPM Field Name | Description |
|---|---|---|
| Producer/Consumer Destination | DESTINATION_NAME_INSTANCE | Destination name from where to send/ receive the messages:<br>• If you are using JNDI, the destination is looked up from the JNDI of the JMS provider.<br>• If you are using Non-JNDI, you need to enter this information.<br>Required. |
| Destination Type | DESTINATION_TYPE_INSTANCE | Destination type. Required. Valid values are:<br>• QUEUE (default)<br>• TOPIC |
| Connection Factory | CONN_FACTORY | Connection Factory name as provided by the JMS Provider. Required. |
| Connection Factory User Name | JMS_FACTORY_USER_NAME | User name for connection factory if the connection factory is secured. Required if connection factory is used. Only displayed if using JNDI. |
| Connection Factory Password | JMS_FACTORY_PASSWORD | Password for connection factory if the connection factory is secured. Required if the connection factory being used is secured. Only displayed if using JNDI. |
| Connection UserName | JMS_CONN_USERNAME | Identifies the Connection User Name. |
| Connection Password | JMS_CONN_PASSWORD | Identifies the Connection Password. |
| Client ID | CLIENT_ID | Identifies the Client ID. Required if using Durable Subscriber. |
| Number of Connections | NO_OF_CONN | Number of connections to be kept in the pool. The pool is lazily initialized. Lazily initialized means that only when a new request comes to create a connection a fresh connection and its associated sessions get created. Required. Valid value is any valid integer. Default is 1. |

| UI Field Name | GPM Field Name | Description |
|---|---|---|
| Number of Sessions | NO_OF_SESSIONS | Number of sessions to be kept in the pool. Required. Valid value is any valid integer. Default is 500. In JMS, sessions are responsible for achieving multi-threaded message delivery. Multi-threaded message delivery is the ability to send/receive messages in parallel at the same time. So ideally the number of sessions should be equal to the number of messages that the you want to send/receive in parallel. (Basically running multiple send/receive business processes in parallel can ensure sending/receiving messages in parallel.) Having this number less than the number of messages that are required to be sent/received in parallel will cause some of the messages to wait while others (number equal to the number of sessions) are being sent/received. |
| Time to Live (per connection in ms) | TIME_TO_LIVE | Amount of time the connection should be kept alive in milliseconds. Required. Valid value is any valid integer. Default is 300000 (milliseconds). If a connection remains idle (not used by either producer or consumer) for this amount of time, whenever a new call comes to create a new connection or use an existing connection from pool, the idle connection is closed and a fresh connection is created and added to the pool. The idle connection will not be closed automatically but only when a new call comes to create/fetch a new connection. |
| Use SSL | SSL_SETTING_ssl_option | Whether to use SSL or not. Required. Valid values are:<br>• SSL_NONE (default)<br>• SSL_MUST |
| CAs Certificate | SSL_SETTING_ca_cert_ids | Select from list of available CA certificates checked in on the server. Required. Only displayed if SSL_MUST was selected for Use SSL. |

| UI Field Name | GPM Field Name | Description |
|---|---|---|
| Use Custom Class Loading | UseCustomClassLoading | Required. |
| | | • When set to false, the third party jars already installed are used. Default is False. |
| | | • When set to true, the user does need not install the third party jars; instead they can provide the path to the jar directory. |
| | | The advantage of using custom class loading include: |
| | | • Eliminates the need to restart the system every time a new vendor jar is added. |
| | | • Handles multiple versions of the same vendor jar. |
| | | You should consider using the custom class loading in following scenarios: |
| | | • The JMS provider being used has jars that conflict with the jars already present in the system and are of a different version. |
| | | • You do not want to add JMS vendor jars in the Sterling Integrator classpath as they might cause some conflicts. |
| | | • You can not afford to restart the system after adding JMS provider jars. |
| | | • You want to run several versions of JMS providers in the system. In that case adding the same vendor jar with multiple versions can cause conflict. |
| | | • You are not sure which jars to keep for the vendor and can not afford to restart the system for every jar combination they add. |
| | | The only limitation is that in certain scenarios with custom class loading customers might have to change the JMS_EXCLUDES list. There is a standard list that has been provided in the documentation for well know vendors. You should review the JMS_EXCLUDES property for proper configuration. Permissions to the jar and directory need to be read for the user that the system is running as. |
| System Certificate | SSL_SETTING_keyCertID | Select from list of available system certificates checked in on the server. Optional. Only displayed if SSL_MUST was selected for Use SSL. |

| UI Field Name | GPM Field Name | Description |
|---|---|---|
| Jar Location | jarLocation | Path and directory to the jars required for custom class loading. Required. This field is only displayed if True was selected for Use Custom Class Loading. In you want to change the jar location or use new jars, you need to release the old custom class loader using the JMS 1.1 Release Connection Session service and then the new class loader needs to be created. If need you more information, see the *JMS 1.1 Release Connection Session service* or the *JMX* documentation. |
| JMS Provider | JmsProviderName | If you are using SSL, select the JMS Provider from the list of providers. Valid values are:<br>• ActiveMQ<br>• WebSphereMQ<br>• TIBCO<br>• WebLogic<br><br>This field is only displayed if SSL_MUST is selected. |
|  | PIN_TO _THREAD | Default is true. Do not change the value of this parameter. This parameter ensures that all the services in the work flow surrounded by service with PIN_TO_THREAD=true indicates the start of the boundary and service with PIN TO THREAD=false is end of that boundary and all services in this boundary will run on the same node and same thread. As a result, all the JMS 1.1 services that are bounded by JMS 1.1 Acquire Connection and Session service (where PIN_TO_THREAD is true) and JMS 1.1 Release Connection and Session service (where PIN_TO_THREAD is marked as false) will always run on the same node and same thread. |

## Parameters Passed From Business Process to Service

The following parameters (GPM field names) are passed from the business process to the JMS 1.1 Acquire Connection and Session service:
• CONN_TYPE
• POOL_NAME
• CONTEXT_FACTORY
• BROKER_URL
• DESTINATION_NAME_INSTANCE
• DESTINATION_TYPE_INSTANCE

- CONN_FACTORY
- JMS_FACTORY_USER_NAME
- JMS_FACTORY_PASSWORD
- JMS_CONN_USERNAME
- JMS_CONN_PASSWORD
- CLIENT_ID
- NO_OF_CONN
- TIME_TO_LIVE
- UseCustomClassLoading
- SSL_SETTING_ca_cert_ids
- SSL_SETTING_keyCertID
- jarLocation
- JmsProvider Name

## Parameters That Must Be Added in BPML

If the Acquire Connection service is already configured you do need not add any parameters to the business process.

## Example Business Processes

The following are two examples of using the JMS1.1 Acquire Connection And Session service.

### Example 1

The following example overrides any parameters in the JMS 1.1 Acquire Connection And Session service. Any parameters set at the business process level will take precedence over those set in the UI.

```
<process name="jms11SendToQueueRegression">
    <sequence>
        <operation name="JMS11AcquireConnSessionService">
            <participant name="JMS11AcquireConnectionAndSessionForQueue" />
             <output message="JMS11AcquireConnSessionServiceInputMessage">
                    <assign to="." from="*"></assign>
                    <assign to="CONN_FACTORY">ConnectionFactoryName
</assign>
                    <assign to="DESTINATION_NAME_INSTANCE">DestinationName
                        </assign>
                    <assign to="DESTINATION_TYPE_INSTANCE">QUEUE</assign>
                    <assign to="NO_OF_CONNS ">1</assign>
                    <assign to="NO_OF_SESSIONS">100</assign>
                    <assign to>
                </output>
                <input message="inmsg">
                    <assign to="." from="*"></assign>
                </input>
        </operation>
        <operation name="JMS11SendMessageService">
            <participant name="JMS11SendMsgRegression" />
             <output message="JMS11SendMessageServiceInputMessage">
                    <assign to="." from="*"></assign>
                    <assign to="MSG_TYPE">TextMessage</assign>
                </output>
                <input message="inmsg">
                    <assign to="." from="*"></assign>
                </input>
```

```
                    </operation>
                    <operation name="JMS11ReleaseConnSessionService">
                            <participant name="JMS11ReleaseConnectionAndSession" />
                            <output message="JMS11ReleaseConnSessionServiceInput
Message">
                                    <assign to="." from="*"></assign>
                            </output>
                            <input message="inmsg">
                                    <assign to="." from="*"></assign>
                            </input>
                    </operation>
            </sequence>
</process>
```

**Example 2**

The following example uses everything from the service which was configured using the UI.

```
<process name="jms11SendToQueueRegression">
      <sequence>
            <operation name="JMS11AcquireConnSessionService">
                    <participant
name="JMS11AcquireConnectionAndSessionForQueue" />
                    <output message="JMS11AcquireConnSessionServiceInputMessage">
                            <assign to="." from="*"></assign>
                     </output>
                     <input message="inmsg">
                            <assign to="." from="*"></assign>
                     </input>
            </operation>
            <operation name="JMS11SendMessageService">
                    <participant name="JMS11SendMsgRegression" />
                    <output message="JMS11SendMessageServiceInputMessage">
                            <assign to="." from="*"></assign>
                            <assign to="MSG_TYPE">TextMessage</assign>
                     </output>
                     <input message="inmsg">
                            <assign to="." from="*"></assign>
                     </input>
            </operation>
            <operation name="JMS11ReleaseConnSessionService">
                    <participant name="JMS11ReleaseConnectionAndSession" />
                    <output
message="JMS11ReleaseConnSessionServiceInputMessage">
                            <assign to="." from="*"></assign>
                     </output>
                      <input message="inmsg">
                            <assign to="." from="*"></assign>
                     </input>
            </operation>
      </sequence>
</process>
```

## Frequently Asked Questions

### Why is my message failing?

If you are using custom class loading in the JMS 1.1, there is a chance that when receiving messages via the Sync Receive Service or the Async Receive Adapter, the replyTo header object in the JMS Message header might be of a class that is not present in the Sterling Integrator dynamic class path.

If this is the situation, then the system is unable to send the message to a destination specified in the replyTo header. You will need to do one of the following:

- Add the jar to the Sterling Integrator dynamic class path rather than using custom class loading.
- Provide a queue/topic name explicitly in the service.

### What does the PIN_TO_THREAD parameter do and should I ever change it?

JMS 1.1 uses a parameter called PIN_TO_THREAD. This parameter ensures that all the services in the work flow surrounded by service with PIN_TO_THREAD=true indicates the start of the boundary and service with PIN_TO_THREAD=false is end of that boundary and all services in this boundary will run on the same node and same thread. As a result, all the JMS 1.1 services that are bounded by JMS 1.1 Acquire Connection and Session service (where PIN_TO_THREAD is true) and JMS 1.1 Release Connection and Session service (where PIN_TO_THREAD is marked as false) will always run on the same node and same thread.

The main reasons for this are around transactionality and utilization of the same session and connection object. So basically between PIN_TO_THREAD=true and PIN_TO_THREAD=false the services will be able to use the same session and connection object. Also, the JMS transactions are governed by session objects. The same session ensures that you can have multiple services using the same session and all of these can be rolled back or committed together.

You should not change this parameter.

# Chapter 122. JMS 1.1 Async Receive Adapter

The JMS 1.1 Async Receive adapter enables a client to receive messages in async mode. A client can register a message consumer against a destination and whenever a message comes to the destination it is passed to the client. Unlike the JMS 1.1 Receive Message service, a client will not need to make a request to the destination for getting the message. The JMS 1.1 Async Receive adapter internally used MessageListener and implements the onMessage() method to receive messages as compared to the receive() call of JMS Receive Message Service.

This adapter internally uses pooling of connections, sessions, consumers, and producers. If you disable this adapter, the connection to the JMS provider is not closed. To close this pool, you will need to run the Business Process suggested in the JMS 1.1 Release Connection Session service. The name of the pool to shutdown is <Name of adapter instance>_<random number>. You should be able to see this in the drop down list for the pools.

Also changing the number of consumers, connection factory, or URL creates another pool with the name <Name of adapter instance>_<different random number> and leaves the original one around until it is either specifically closed for the node is reset. If you want to edit the adapter parameters, you should disable the adapter first, shutdown the pool, and then enable the adapter. If the older pool is not shutdown and the parameters are changed, a new pool will get created and the older pool remains enabled though it will not be used. The older pool is still available, until you close it or reset the node. To free the connections the older pool is using, it is recommended that the older pool is shut down after disabling the adapter and before editing the parameters and starting the adapter instance.

The following table provides an overview of the JMS 1.1 Async Receive adapter:

| System Name | JMS 1.1 Async Receive adapter |
|---|---|
| Graphical Process Modeler (GPM) categories) | None |
| Description | The JMS 1.1 Async Receive adapter enables a client to receive messages in async mode. A client can register a message consumer against a destination and whenever a message comes to the destination it is passed to the client. Unlike JMS 1.1 Receive Message service, a client will not need to make a request to the destination to receive the message. |
| Business usage | You can receive messages from a trading partner using a JMS 1.1 messaging queue or topic in async mode. |
| Usage example | Based on low inventory of a certain item, your company sends a JMS message to a queue stating that more inventory that needs to be sent. You configure an instance of JMS 1.1 Async Receive adapter which listens to this queue. As soon as a message is received on the queue, the JMS 1.1 Async Receive adapter triggers a business process responsible for providing the necessary inventory to your company. |
| Preconfigured? | No |

| Requires third party files? | The JMS provider jar must be installed on the server using install3rdParty script. Before installing the jar file, you need to know:<br>• Remote JMS server context factory filename<br>• URL (IP address and port)<br>• Connection factory file name and location<br>• User name and password<br>• Public key for the CA or trusted certificates (if used) from the trading partner for access to their JMS server or system certificate (if used) |
|---|---|
| Platform availability | All supported platforms |
| Related services | Related services:<br>• JMS 1.1 Acquire Connection and Session service<br>• JMS 1.1 Release Connection and Session service<br>• JMS 1.1 Send Message service<br>• JMS 1.1 Request Reply service<br>• JMS 1.1 Receive Message service |
| Application requirements | It depends on how you plan to use JMS. Creation of a JMS connection factory and destinations will be required if you want to use the JNDI approach for JMS. |
| Initiates business processes? | Yes. When a message is received from the specified JMS provider, the adapter bootstraps the business process you specified during adapter configuration. This adapter cannot be used in a business process. It should always be used as a listening adapter and never in a business process. |
| Invocation | Event driven. Triggers a business process for every JMS message received. |
| Business process context considerations | None |
| Returned status values | None |
| Restrictions | Default |
| Persistence level | None |
| Testing considerations | None |

## Configuring the JMS 1.1 Async Receive Adapter in the Admin UI Console

Use the field definitions in the following table to create a new configuration of the JMS 1.1 Async Receive adapter.

| UI Field Name | Description |
|---|---|
| Name | Unique name for the service configuration. Required. |
| Description | Description for the service configuration, for reference purposes. Required. |

| UI Field Name | Description |
|---|---|
| Select a Group | Select one of the options: <br><br>• None – Do not include the configuration in a service group at this time. <br>• 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.) <br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |
| Connection Type | Defines whether to use JNDI or not. Required. Valid values are: <br><br>• Using JNDI (Default) - Use JNDI when the JMS administered objects (such as, Connection Factory and Destination) needs to be looked up from the JNDI of the JMS provider. This can be useful if the provider wants to control the creation of these administered objects rather than JMS clients and wants to share the same across all the clients. Some vendors (WebLogic and JBoss) only support JNDI mode. <br>• Using Non -JNDI - If the client wants to create the administered objects in the client code. Several JMS vendors do not support Non-JNDI. <br><br>If you need additional information to determine which connection type is supported by your vendor, consult the vendor documentation. |
| Context Factory | Context Factory name as provided by the JMS Provider. Required. Only displayed if using JNDI. |
| URL | URL needed to connect with the JMS provider. Required. |
| Connection Factory | Connection Factory name as provided by the JMS Provider. Required. |
| Receive Destination | Destination name from where to receive the messages. Required. |
| Connection Factory User Name | User name for JNDI look up, if the JNDI look up is secured. This field is only displayed for JNDI. |
| Connection Factory Password | Password for JNDI look up, if the JNDI look up is secured. This field is only displayed for JNDI. |
| Connection User Name | Identifies the Connection User Name. |
| Connection Password | Identifies the Connection Password. |
| Message Selector | Message selector in case the customer wants messages matching only the given selector to be picked. |
| Destination Type | Destination type. Required. Valid values are: <br><br>• QUEUE (default) <br>• TOPIC |

| UI Field Name | Description |
|---|---|
| Time to Live (per connection in ms) | Amount of time the connection should be kept alive in milliseconds. Required. Valid value is any valid integer. Default is 300000 (milliseconds).If a connection remains idle (not used by either producer or consumer) for this amount of time, whenever a new call comes to create a new connection or use an existing connection from pool, the idle connection is closed and a fresh connection is created and added to the pool. The idle connection will not be closed automatically but only when a new call comes to create/fetch a new connection. |
| Concurrent Consumers | Concurrent number of consumers on this given destination at any given time This parameter should be adjusted for performance. Required. Valid value is an integer. By default for FIFO mode or when using TOPICS this value is always set to 1. |
| Is Session Transacted | Whether the session that receives this message should be transacted or not. If transacted, it ensures that the message is not lost even if the application crashes. Required. Valid values are:<br>• Yes<br>• No (default) |
| Success Business Process | The business process that should be bootstrapped when a message is received. Required. Select from the list of business processes.If you need more information, see the Process Execution Mode and Is Session Transacted parameter descriptions. |
| Process Execution Mode | Defines how the business process is queued at run time. Select the process execution mode from the list. Valid values are:<br>• Default (default) - Async mode. On receiving a message a Business Process is kicked off in a separate thread. This ensures that the adapter is free to receive another message without waiting for the business process to finish. Using transactions here ensures that if system fails to persist the message then it is returned back to the provider. If the business process fails, then the message is NOT returned back.<br>• FIFO - First In First Out (FIFO) mode ensures that the bootstrapped business process runs in FIFO mode. It can be considered as an async mode that guarantees the ordering of bootstrapped business processes. FIFO mode can be used where order of the bootstrapped business process should be same as received message. FIFO provides the ability to kick off a business process in a separate thread, but ensures that the kicked off business processes follows the correct order. As a result, you should be careful when selecting your transaction strategy with FIFO.<br>• Non Queued - Sync mode. On receiving a message a Business Process is kicked off in the same thread. This ensures that the adapter is blocked, new messages are not received until the business process has completed. Using transactions here ensures that if the business process fails, then the message is returned back. |

| UI Field Name | Description |
|---|---|
| Document Storage Type | Document storage type for received messages. Valid values are:<br>• System Default (default)<br>• Database<br>• File System |
| Document Filename | Filename for the data that is received. You can use a generator placeholder, %^, which generates a sequence in the form of <nodename>_yyyymmddhhmmsslll.It is important that the document file name is a unique name. If the same document file name is used, it can cause potential data loss scenario as the file could potentially be overwritten. |
| Use Durable Subscriber | Ensures durable subscription to a topic. Optional. Only valid when Destination Type is TOPIC. Valid values are:<br>• Yes<br>• No (default) |
| Client ID | Client ID is set in the connection when using durable subscription. Optional. Only valid when Destination Type is TOPIC. |
| Durable Subscriber Name | Name used when using durable subscription. Only valid when the Destination Type is TOPIC. Required, when Durable Subscription value is Yes. |
| User | User name used when bootstrapping the business process. Required. Select the user name from the list. |
| Connection retry attempts | Number of retry attempts to connect to a JMS server if connection is lost. Optional. |
| Delay between retries | Delay between each retry attempt. Optional. Default value is 0, retries will not happen. |
| Notification Business Process | Notification business process to bootstrap if unable to connect to server after the number of retries. Optional. |
| Use SSL | Whether to use SSL or not. Required. Valid values are:<br>• SSL_NONE (default)<br>• SSL_MUST |
| SSL SETTING ca cert ids | Select from list of available CA certificates checked in on the server. Required. Only displayed if SSL_MUST was selected. |
| SSL SETTING key certID | Select from list of available system certificates checked in on the server. Optional. |
| JMS Provider | If you are using SSL, select the JMS Provider from the list of providers. Valid values are:<br>• ActiveMQ<br>• WebSphereMQ<br>• TIBCO<br>• WebLogic |

| UI Field Name | Description |
|---|---|
| jar Location | Path and directory to the jars required for custom class loading. Required. This field is only displayed if True was selected for Use Custom Class Loading.In you want to change the jar location or use new jars, you need to release the old custom class loader using the JMS 1.1 Release Connection Session service and then the new class loader needs to be created.If need you more information, see the *JMS 1.1 Release Connection Session service* or the *JMX* documentation. |
| Use Custom Class Loading | Required.<br>• When set to false (no), the third party jars already installed are used. Default is False.<br>• When set to true (yes), the user does need not install the third party jars; instead they can provide the path to the jar directory.<br><br>The advantage of using custom class loading include:<br>• Eliminates the need to restart the system every time a new vendor jar is added.<br>• Handles multiple versions of the same vendor jar.<br><br>You should consider using the custom class loading in following scenarios:<br>• The JMS provider being used has jars that conflict with the jars already present in the system and are of a different version.<br>• You do not want to add JMS vendor jars in the Sterling Integrator classpath as they might cause some conflicts.<br>• You can not afford to restart the system after adding JMS provider jars.<br>• You want to run several versions of JMS providers in the system. In that case adding the same vendor jar with multiple versions can cause conflict.<br>• You are not sure which jars to keep for the vendor and can not afford to restart the system for every jar combination they add.<br><br>The only limitation is that in certain scenarios with custom class loading customers might have to change the JMS_EXCLUDES list. There is a standard list that has been provided in the documentation for well know vendors. You should review the JMS_EXCLUDES property for proper configuration.Permissions to the jar and directory need to be read for the user that the system is running as. |
| System Certificate | System Certificate Name. Required when SSL_MUST is selected for Use SSL. |
| CA Certificate | CA Certificate Name. Required when SSL_MUST is selected for Use SSL. |

## Frequently Asked Questions

## Why is my message failing?

If you are using Custom Class Loading, there is a chance that when receiving messages via the JMS 1.1 Sync Receive Service or the JMS 1.1 Async Receive Adapter, that the replyTo header object in the JMS Message header might be a class that is not present in the Sterling Integrator dynamic class path.

If this is the situation, the system is unable to send the message to a destination specified in the replyTo header. To correct this, you will need to do one of the following:

- Add the jar to the Sterling Integrator dynamic class path rather than using custom class loading.
- Provide a queue/topic name explicitly in the service.

## What happens when the JMS provider goes down?

For more information about JMS 1.1 Pools, see the JMS 1.1 Pool documentation topic.

## Do I need to shutdown the pool every time I shutdown my adapter?

Yes. If you plan to change any of the pool parameters, you need to:

1. Shutdown the adapter.
2. Shutdown the pool.
3. Edit the parameters.
4. Re-enable the adapter.

Failure to do so may result in having multiple pools with same name.

## How is the pool affected when I edit my adapter rather than shutting it down?

If you edit your adapter instance before shutting it down and end up changing one of the pool parameters, you might end up with multiple pools with the same name.

## How do I use transactions in JMS 1.1 Async Receive adapter?

Transactions can be enabled in JMS 1.1 Async Receive Adapter by setting the IS SESSION TRANSACTED parameter to YES.

There are two scenarios where transactions in async mode can be useful:

- When the bootstrap mode is Default or Async: If the business process is kicked off in the async mode as soon as a message is received. But before kicking off the business process the payload is stored in database. The transaction is committed as soon as the persistence to database is successful. If the database is down, then the session is rolled back and the message is returned. Failure of bootstrapped business process does not ensure that the message is returned to the destination.
- When the bootstrap mode is Sync: Once a message is received the business process is kicked off in sync mode (until the time the business process has completed executing the new message is not picked up). The session is

committed only if the bootstrapped business process has successfully finished. Failure in the business process results in returning the messaged back to the destination.

## Why do I keep receiving the same message when using transactions in JMS 1.1 Async Receive adapter?

When using transactions, if a transaction is not committed (because of any failures) then the same message is returned. Depending on the redelivery policy of the destination and the message, the same message might be returned to this destination. You might receive the same message again. You should configure the redelivery mechanism at both destination and message level.

Customers can check to see if the received message is delivered for the first time or whether it is being redelivered, because the delivery failed first time by checking the JMSRedelivered message header of the received message. This value can either be true or false depending on whether the message is being redelivered or not. Depending on this flag, the bootstrapped business process may decide to choose or carry or ignore certain steps.

## How do I set up a durable subscription in JMS 1.1 Async Receive adapter?

Setting up a durable subscription for the JMS 1.1 Async Receive adapter is similar to the setting up the JMS 1.1 Receive Message service. The first time the adapter is started the durable subscription is registered. You will need to restart the adapter to start getting messages using durable subscription.

## How can I unsubscribe from a durable subscription in case of JMS 1.1 Async Receive Adapter?

Shutting down the adapter does not unsubscribe the durable subscription. To unsubscribe from durable subscription, run the following work flow:

Start >JMS 1.1 Release Connection and Session service > Stop

The parameters that need to be set in the JMS 1.1 Release Connection and Session service are:
- RELEASE ACTION should be set as UNSUBSCRIBE DURABLE.
- RELEASE DURABLE SUBSRIBER should be provided with the durable subscriber name that was provided in the JMS 1.1 Receive Message Service.
- POOL NAME that was provided in the JMS 1.1 Acquire Connection and Session service when the JMS 1.1 Receive Message service was used.

# Chapter 123. JMS 1.1 Receive Message Service

The JMS 1.1 Receive Message service creates a JMS message consumer and is responsible for receiving messages from the specified destination. The following table provides an overview of the JMS 1.1 Receive Message service:

| | |
|---|---|
| System Name | JMS 1.1 Receive Message service |
| Graphical Process Modeler (GPM) categories) | All Services |
| Description | JMS 1.1 Receive Message service receives messages synchronously from the remote JMS destination (Queue/Topic). |
| Business usage | JMS 1.1 Receive Message service receives messages synchronously from the remote JMS destination (Queue/Topic). |
| Usage example | Use this service to receive messages from remote Queues or Topics. |
| Preconfigured? | Yes |
| Requires third party files? | The JMS provider jar must be provided to the system, using one of the following methods:<br><br>• Install using the install3rdParty script<br><br>• Load using the custom classloader<br><br>For JBoss and WebLogic, you must use the custom classloader as these vendor jars contain classes that already exist in system classloader. For example, if you need to communicate to an external WebLogic JMS server, you need the jar file that includes the weblogic.jndi.WLInitialContextFactory class. You would obtain the jar file from the corresponding vendor or your trading partner. |
| Platform availability | All supported platforms |
| Related services | This adapter can be used in conjunction with the JMS 1.1 services that are included in business processes:<br><br>• JMS 1.1 Acquire Connection and Session service and JMS 1.1 Release Connection and Session service – these services are used to start and stop a JMS session within a business process. They must be used as a pair in a business process.<br><br>• JMS 1.1 Send Message service – used to send messages to a JMS topic or queue. |
| Application requirements | None |
| Initiates business processes? | No |
| Invocation | None |
| Business process context considerations | None |
| Returned status values | None |
| Restrictions | None |
| Persistence level | Minimum |
| Testing considerations | None |

# Configuring the JMS 1.1 Receive Message Service

You must configure this services using the GPM.

| UI Field | 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:<br><br>• None – Do not include the configuration in a service group at this time.<br><br>• 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.)<br><br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |
| | Config | Select the configuration from the list. |
| | BATCH_MODE | When batch mode is specified, it indicates the service is in the batch receiving mode. Valid value is any valid integer. Default value is 1. For example, the service will receive 4 messages when this parameter is set to an integer 4. If there are only 3 messages to receive, it will wait for the fourth one until the time limit and then returns with 3 messages. |
| | DESTINATION_NAME _WFC | Specifies the Destination Name in the Receive Service. If Acquire Connection And Session Service are specified, the value specified in business process will override this parameter. |
| | DESTINATION_TYPE_ WFC | Specifies the Destination Type (QUEUE/ TOPIC) in the Receive Service. If Acquire Connection And Session Service are specified, the value specified in business process will override this parameter. |
| | DURABLE_SUBSCRIBER_NAME | Specifies the name of the durable subscription on the subscriber. Required if IS_DURABLE_SUBSCRIBER is set to Yes. |

| UI Field | GPM Field | Description |
|---|---|---|
| | IS_DURABLE_SUBSCRIBER | Whether a subscriber to be created should be durable or not. This is only applicable to the scenarios where TOPIC is used as DESTINATION_TYPE. Values are:<br>• YES<br>• NO |
| | MESSAGE _SELECTOR | Specifies the selector filter on the consumer. Example values are:<br><br>JMSCorrelationID= 'ABC'JMSPriority=6JMSPriority=6 AND JMSCorrelationID= 'abc' JMSPriority=6 OR JMSCorrelationID='abc' |
| | OUTPUT_FILE_NAME | User specified file name for the data that gets received from the destination. If you are receiving documents in batch mode or from the Async Adapter, you can use %^ as the output file name. When using the %^, the output file name is assigned in the format <nodename>_yyyymmddhhmmsslll. |
| | RECEIVE_ TIME_LIMIT | Specifies the time limit (in milliseconds) for receiving the message. Valid value is any valid integer. Default is 300000 (milliseconds). |

## Business Process Examples

### Example 1

The following example illustrates how the JMS 1.1 Receive Message service can be used in a business process. This example does not override any parameters in the Receive Service. It uses JMS 1.1 Acquire Connection And Session For Queue service which is UI configured and preset with all the parameters.

```
<process name="jms11ReceiveFromQueueRegression">
   <sequence>
    <operation name="JMS11AcquireConnSessionService">
      <participant name="JMS11AcquireConnectionAndSessionForQueue"/>
      <output message="JMS11AcquireConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMS11ReceiveMessageService">
      <participant name="JMS11ReceiveMsg"/>
      <output message="JMS11RecvMessageServiceInputMessage">
        <assign to="." from="*"></assign>
            </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
```

```
      </operation>
      <operation name="JMS11ReleaseConnSessionService">
        <participant name="JMS11ReleaseConnectionAndSession"/>
        <output message="JMS11ReleaseConnSessionServiceInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
    </sequence>
</process>
```

## Example 2

This example shows using the Receive service by specifying the Destination Name and Type and also setting other parameters like batch mode and receive time limit. When this business process is executed, four messages are received from the testDestination_Topic.

```
<process name="jms11SendToQueueRegression">
  <sequence>
    <operation name="JMS11AcquireConnSessionService">
      <participant name="JMS11AcquireConnectionAndSessionForQueue"/>
      <output message="JMS11AcquireConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
<operation name="JMSRecieveMessageService">
      <participant name="JMS11ReceiveMsg"/>
      <output message="JMSRecieveMessageServiceInputMessage">
        <assign to="." from="*"></assign>
        <assign to="BATCH_MODE">4</assign>
        <assign to="DESTINATION_NAME_WFC">testDestination_Topic</assign>
        <assign to="DESTINATION_TYPE_WFC">Topic</assign>
        <assign to="RECEIVE_TIME_LIMIT">40000</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMS11ReleaseConnSessionService">
      <participant name="JMS11ReleaseConnectionAndSession"/>
      <output message="JMS11ReleaseConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

## Example 3

This example demonstrates the use of Durable Subscriber and the Message Selector. Here the durable subscriber testDurable gets created and it receives messages which have a Correlation ID equals to "ABC". While using the durable subscriber, the client ID must be set in JMS 1.1 Acquire Connection service.

```
<process name="jms11BatchSend">
  <sequence>
    <operation name="JMS11AcquireConnSessionService">
```

```
            <participant name="JMS11AcquireConnectionAndSessionForQueue"/>
            <output message="JMS11AcquireConnSessionServiceInputMessage">
            <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
              <assign to="." from="*"></assign>
            </input>
        </operation>
 <operation name="JMSRecieveMessageService">
            <participant name="JMS11ReceiveMsg"/>
            <output message="JMSRecieveMessageServiceInputMessage">
              <assign to="." from="*"></assign>
               <assign to="DESTINATION_NAME_WFC">testDestination_Topic</assign>
              <assign to="DESTINATION_TYPE_WFC">Topic</assign>
        <assign to="DURABLE_SUBSCRIBER_NAME">testDurable</assign>
              <assign to="IS_DURABLE_SUBSCRIBER">Yes</assign>
              <assign to="MESSAGE_SELECTOR">JMSCorrelationID='ABC'</assign>
                    </output>
            <input message="inmsg">
              <assign to="." from="*"></assign>
            </input>
        </operation>
        <operation name="JMS11ReleaseConnSessionService">
          <participant name="JMS11ReleaseConnectionAndSession"/>
          <output message="JMS11ReleaseConnSessionServiceInputMessage">
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="." from="*"></assign>
          </input>
        </operation>
      </sequence>
</process>
```

## Example 4

The following example shows how to receive a message using transactions and
rollback in case transaction fails. You can also use other services that can use
transactions like Lightweight JDBC in combination with the below and make them
commit or rollback together. For example in the below mentioned workflow,
LightweightJDBC service instance can be added before or after JMS 1.1 Receive
Message service to ensure that if both receiving of message and successful
execution of database query happens then only the transaction gets committed else
the transaction is rolled back.

```
<process name="default">
   <sequence>
      <operation name="Begin Transaction Service">
          <participant name="BeginTransactionService" />
          <output message="BeginTransactionServiceTypeInputMessage">
                      <assign to="." from="*"></assign>
                      <assign to="START_TRANSACTION">TRUE</assign>
                      <assign to="DISTRIBUTED">TRUE</assign>
                      <assign to="ON_FAULT">ROLLBACK</assign>
                  </output>
                  <input message="inmsg">
                      <assign to="." from="*"></assign>
                  </input>
      </operation>
    <operation name="JMS 1.1 Acquire Connection and Session Service">
        <participant name="JMS11AcquireConnectionAndSessionForQueue
Regression"/>
        <participant name="JMS11AcquireConnectionAndSessionForQueue
Regression"/>
                <output message="JMS11AcquireConnSessionServiceInputMessage">
                      <assign to="." from="*"></assign>
```

```
                </output>
                <input message="inmsg">
                        <assign to="." from="*"></assign>
                </input>
            </operation>
         <operation name="JMS 1.1 Recieve Message Service">
                <participant name="JMS1.1ReceiveMessageService" />
                    <output message="JMS11RecieveMessageServiceInputMessage">
                            <assign to="." from="*"></assign>
                    </output>
                    <input message="inmsg">
                            <assign to="." from="*"></assign>
                    </input>
            </operation>
            <operation name="End Transaction Service">
                    <participant name="EndTransactionService" />
                    <output message="EndTransactionServiceTypeInputMessage">
                            <assign to="." from="*"></assign>
                            <assign to="ROLLBACK_TRANSACTION">FALSE</assign>
                            <assign to="END_TRANSACTION">TRUE</assign>
                    </output>
                    <input message="inmsg">
                            <assign to="." from="*"></assign>
                    </input>
            </operation>
          <operation name="JMS 1.1 Release Connection and Session Service">
                    <participant name="JMS11ReleaseConnectionAndSessionRegression"/>
                    <output message="JMS11ReleaseConnSessionServiceInputMessage">
                            <assign to="." from="*"></assign>
                    </output>
                      <input message="inmsg">
                            <assign to="." from="*"></assign>
                      </input>
            </operation>
        </sequence>
</process>
```

If needed, refer to the Begin and End JTA transaction services for more details.

## Example 5

The following example shows how a message received can be sent to a destination
present in the ReplyTo destination of the header. The example assumes that no
destination has been specified in the JMS 1.1 Acquire Connection and Session
service or JMS 1.1 Send Message Service or JMS 1.1 Receive Message Service. Also
the received message has a destination present in the replyTo header of the
received message.

```
<process name="default">
    <sequence>
        <operation name="JMS 1.1 Acquire Connection and Session Service">
            <participant name="JMS11AcquireConnectionAndSessionFor
                        QueueRegression" />
            output message="JMS11AcquireConnSessionServiceInputMessage">
                        <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
                        <assign to="." from="*"></assign>
            </input>
        </operation>
        <operation name="JMS 1.1 Recieve Message Service">
                <participant name="JMS1.1ReceiveMessageService" />
                <output message="JMS11RecieveMessageServiceInputMessage">
                        <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
```

```
                    <assign to="." from="*"></assign>
            </input>
        </operation>
        <operation name="JMS 1.1 Send Message Service">
            <participant name="JMS1.1SendMessageService" />
            <output message="JMS11SendMessageServiceInputMessage">
                <assign to="." from="*"></assign>
                    <assign to="MSG_TYPE">TextMessage</assign>
            </output>
            <input message="inmsg">
                    <assign to="." from="*"></assign>
            </input>
        </operation>
        <operation name="JMS 1.1 Release Connection and Session Service">
            <participant name="JMS11ReleaseConnectionAndSession
                        Regression" />
            <output message="JMS11ReleaseConnSessionServiceInputMessage">
                    <assign to="." from="*"></assign>
            </output>
             <input message="inmsg">
                    <assign to="." from="*"></assign>
            </input>
        </operation>
    </sequence>
</process>
```

## Frequently Asked Questions

## What if I do not receive a message within a given time?

If the message is not received by the JMS 1.1 receive message service, within the specified time out, the business process completes successfully, but no document will be present.

## How do I receive custom JMS Message properties?

All the custom message properties and headers present in the received message are appended to the primary document in the following format.

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
<JMS>
    <Document SCIObjectID="server:node1:127f5dcff5e:2169">
      <JMSCustomProperty>CustomValue</ JMSCustomProperty >
    </Document>
  </JMS>
</ProcessData>
```

## What are the various levels where I can set destination override?

For the JMS 1.1 Receive Message service, the destination can be overridden at acquire as well as receive level.

## How do I set up a durable subscription?

A durable subscription is only valid for topics (queues are always durable). To set up a durable subscription you need to provide the CLIENT ID in the JMS 1.1 Acquire Connection and Session service. You also need to specify the IS DURABLE SUBSCRIBER parameter and DURABLE SUBSCRIBER NAME in the JMS 1.1 Receive Message service.

The first time this business process runs, the durable subscription is set up, subsequent business processes run should be able pick up messages in the durable mode. To unsubscribe from this subscription you will need to manually shutdown the durable subscription.

## How can I unsubscribe from a durable subscription?

To unsubscribe from the durable subscription, run the following work flow:

Start >JMS 1.1 Release Connection and Session service > Stop

The parameters that need to be set in the JMS 1.1 Release Connection and Session service are:
- RELEASE ACTION should be set as UNSUBSCRIBE DURABLE.
- RELEASE DURABLE SUBSRIBER should be provided with the durable subscriber name that was provided in the JMS 1.1 Receive Message Service.
- POOL NAME that was provided in the JMS 1.1 Acquire Connection and Session service when the JMS 1.1 Receive Message service was used.

## What happens when the JMS provider goes down?

Since the JMS 1.1 Acquire Connection And Session service is responsible for getting the connection and session, the work flow will fail with an exception message in the creating connection.

## How do I use transactions in JMS 1.1 Receive service?

The same way you use transactions in the JMS 1.1 Send Message service.

## How do I receive multiple messages?

You can specify the parameter BATCH_LIMIT to specify the number of messages you want to receive. The service will be marked complete only after receiving that many messages or time out, whatever comes first.

## What if I receive messages less than the specified batch number? Will I still be able to process the remaining messages and how will I figure out how many messages I am short of ?

Basically it is like calling the receive in a loop with a timeout. You may or may not get a message within timeout. If you do, then we take the message and add a document tag with a number to the process data else not. You can find the number of document tags in the process data and see how many messages you are short.

## Is the time out parameter per message or overall time out when receiving multiple messages?

Time out is per message.

For example, if you have specified a batch size of 5 and have only 3 messages to receive, and the timeout is 1 minute. You start receiving first message, wait for 1 minute before giving up, then start with second wait another minute and so on.

## What happens if the RECEIVE_TIME_LIMIT expires?

If message is not received within the given timeout by the JMS 1.1 receive message service the business process completes successfully but no document will be present.

The lack of a Document is the indicator if no message is received within a given time. You can automate this by checking if the Document Tag is present in the process data and has a value assigned to it. For example, process data when message is received:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
<JMS>
    <Document SCIObjectID="server:node1:127f5dcff5e:2169">
    </Document>
  </JMS>
</ProcessData>
Process data when message is received:
<?xml version="1.0" encoding="UTF-8"?>
<ProcessData>
<JMS>
  </JMS>
</ProcessData>
```

# Chapter 124. JMS 1.1 Release Connection and Session Service

The JMS 1.1 Release Connection and Session service returns a JMS connection and session back to the pool. The following table provides an overview of the JMS 1.1 Release Connection and Session service:

| | |
|---|---|
| System Name | JMS 1.1 Release Connection and Session service |
| Graphical Process Modeler (GPM) categories) | All Services |
| Description | The JMS 1.1 Release Connection and Session service returns a JMS connection and session back to the pool. |
| Business usage | The service can be used in a normal JMS 1.1 business process with the mode set as NONE. In this scenario, the JMS 1.1 Release Connection and Session service is used to release connections and sessions used to pool. The service can also be used individually to shut down pools and to unsubscribe durable subscribers. |
| Usage example | Normal usage includes using this service in association with any JMS 1.1 business process. The service can also be used to shutdown pools and release durable subscribers. |
| Preconfigured? | Yes. |
| Requires third party files? | The JMS provider jar must be provided to the system, using one of the following methods:<br>• Install using the install3rdParty script<br>• Load using the custom classloader<br><br>For JBoss and WebLogic, you must use the custom classloader as these vendor jars contain classes that already exist in system classloader. For example, if you need to communicate to an external WebLogic JMS server, you need the jar file that includes the weblogic.jndi.WLInitialContextFactory class. You would obtain the jar file from the corresponding vendor or your trading partner. |
| Platform availability | All supported platforms |
| Related services | This service is used with the JMS 1.1 Acquire Connection and Session service and JMS 1.1 Send and Receive service. The JMS 1.1 connection and session required for Request Reply service are created by the JMS 1.1 Acquire Connection and Session service. |
| Application requirements | None |
| Initiates business processes? | No |
| Invocation | Invoked as a step in a business process. |
| Business process context considerations | None |
| Returned status values | None |
| Restrictions | None |
| Persistence level | Minimum |

| | | |
|---|---|---|
| Testing considerations | | Set the debug level ON from the Log screen in the system for Log.jms. |

## Configuring the JMS 1.1 Release Connection and Session Service

You must use the GPM to configure this service.

| UI Field | 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:<br>• None – Do not include the configuration in a service group at this time.<br>• 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.)<br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |
| | Config | Select the name of the service configuration from the list. |
| | PIN_TO_THREAD | Default is False. Do not change this value. |
| | POOL_LIST | When the service is used to shut down a pool or unsubscribe a durable subscriber, POOL_LIST defines the name of the pool to be shut down. Select the pool name from the list. No default value. If all of the pools need to be shut down, select ALL_JMS_POOLS. Required when the mode is SHUTDOWN_POOL or UNSUBSCRIBE_DURABLE. |

| UI Field | GPM Field | Description |
|---|---|---|
| | RELEASE_ACTION | Indicates whether the service is being used to release a connection and session, or to shut down a pool. Required. Valid values are:<br>• NONE – Release connection and session to the pool<br>• SHUTDOWN_POOL – use to shut down the pool<br>• SHUTDOWN_RR – use to shut down RR<br>• UNSUBSCRIBE_DURABLE - use to unsubscribe a durable subscriber |
| | RELEASE_DURABLE_SUBSCRIBER△ | Indicates the name of the durable subscriber. |
| | RR_LIST | Lists all of the request reply instances created for permanent destination. |
| | SEGMENTED_POOL_LIST | Name of the segmented pool needed to shut down or unsubscribe a durable subscriber in case the service is being used to shut down a pool. Select the pool name from the list. If all of the pools need to be shutdown, select ALL_JMS_POOLS. |

## Business Process Examples

Pool names need to have a unique number added to them in addition to the name specified in the pool name. This number is changed when the pool is edited in acquire connection session service. As a best practice change the pool name when editing or updating a pool to avoid confusion.

## Example 1

The following example illustrates how the JMS 1.1 Release Connection and Session service can be used in a business process to shut down a pool:

```
process name="default">
  <operation name="JMS11ReleaseConnSessionService">
    <participant name="testRegressionsReleaseConnSessionService"/>
    <output message="JMS11ReleaseConnSessionServiceInputMessage">
      <assign to="." from="*"></assign>
      <assign to="POOL_LIST">JMS11RegressionTestPoolForQueue_-484870860
            </assign>
      <assign to="RELEASE_ACTION">SHUTDOWN_POOL</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

## Example 2

The following example illustrates how the JMS 1.1 Release Connection and Session service can be used in a business process to shut down a reply request:

```
<process name="default">
  <operation name="JMS 1.1 Release Connection and Session Service">
    <participant name="testGPMSendRelease"/>
    <output message="JMS11ReleaseConnSessionServiceInputMessage">
      <assign to="." from="*"></assign>
      <assign to="RR_LIST">rr_to_shutdown</assign>
      <assign to="RELEASE_ACTION">SHUTDOWN_RR</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

## Example 3

The following example illustrates how the JMS 1.1 Release Connection and Session service can be used in a business process to shut down a Segmented Pool or refresh the classloader for a pool:

```
<process name="default">
  <operation name="JMS 1.1 Release Connection and Session Service">
    <participant name="testGPMSendRelease"/>
    <output message="JMS11ReleaseConnSessionServiceInputMessage">
      <assign to="." from="*"></assign>
      <assign to="RELEASE_ACTION">SHUTDOWN_POOL</assign>
<assign to="SEGMENTED_POOL_LIST">segmentedPool</assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

## Example 4

The following example illustrates how the JMS 1.1 Release Connection and Session service can be used in a business process to unsubscribe a durable subscriber:

```
<process name="default">
     <operation name="JMS 1.1 Release Connection and Session Service">
          <participant name="testGPMSendRelease" />
          <output message="JMS11ReleaseConnSessionServiceInputMessage">
                <assign to="." from="*"</assign>><assign to="." from="*">
                     </assign>
                <assign to="POOL_LIST">somePoolName</assign>
                <assign to="RELEASE_ACTION">UNSUBSCRIBE_DURABLE</assign>
                <assign to="RELEASE_DURABLE_SUBSCRIBER">durableSubscriberNam
                     </assign>e<assign to="RELEASE_DURABLE_SUBSCRIBER">
                              durableSubscriberName</assign>
          </output>
          <input message="inmsg">
                <assign to="." from="*"></assign>
          </input>
     </operation>
</process>
```

## Frequently Asked Questions

### What are the various uses of JMS 1.1 Release Connection and Session service?

JMS 1.1 Release Connection and Session service serves the following purposes:

- When used with other JMS services, and RELEASE ACTION is NONE, the service is used to return the connection and sessions created by the acquire service back to the pool.
- When used alone, and the RELEASE ACTION is SHUTDOWN POOL, the service can be used to shutdown the JMS pool or can be used to SHUTDOWN a segmented pool so that new jars can be used in place of the old ones for Custom Class Loading.
- When used alone, and the RELEASE ACTION is SHUTDOWN RR, the service can be used to shutdown the async receiver created by the JMS 1.1 Request Reply Service.
- When used alone, and the RELEASE ACTION is UNSUBSCRIBE DURABLE SUBSCRIBER, the service can be used to unsubscribe the durable subscriber.

### What is the difference between the POOL LIST and the SEGMENTED POOL LIST parameters in JMS 1.1 Release Connection and Session service?

All the JMS pools created using the dynamic classpath (classpath that Sterling Integrator uses by default) are listed in POOL LIST. All the JMS pools created using Custom Class Loading are listed in the SEGMENTED POOL LIST.

The advantage of having a SEGMENTED POOL LIST is that if the customer plans to change the jars in the specified folder, all that needs to be done is (if it is an adapter then the adapter needs to be shutdown first) to run the JMS 1.1 Release Connection and Session service, select the action as SHUTDOWN POOL and select the pool from SEGMENTED POOL LIST. This shutdowns the pool along with clearing the Custom Class Loader. Now the jars can be changed and a new class loader will be used to pick up these jars.

# Chapter 125. JMS 1.1 Request Reply Service

The JMS 1.1 Request Reply service is used in scenarios where response from user matters to the sender and depending on this response an action needs to be taken by the sender. The following table provides an overview of the JMS 1.1 Request Reply service:

| | |
|---|---|
| System Name | JMS 1.1 Request Reply service |
| Graphical Process Modeler (GPM) categories) | Services |
| Description | This service is used in scenarios where response from user matters to the sender and depending on this response an action needs to be taken by the sender. |
| Business usage | This service is used in scenarios where response from user matters to the sender and depending on this response an action needs to be taken by the sender. |
| Usage example | For example, consider credit card verification. In this example, the payment should be blocked until the credit card authentication happens. The user sends a request to verify credit card credentials to the bank. The bank verifies the credentials and sends a reply back and then the flow proceeds. No other steps are carried out until bank has replied. |
| Preconfigured? | Yes |
| Requires third party files? | The JMS provider jar must be provided to the system, using one of the following methods:<br>• Install using the install3rdParty script<br>• Load using the custom classloader<br><br>For JBoss and WebLogic, you must use the custom classloader as these vendor jars contain classes that already exist in system classloader. |
| Platform availability | All supported platforms |
| Related services | This service is used with JMS 1.1 Acquire Connection and Session service and JMS 1.1 Release Connection and Session service. The JMS connection and session required for the Request Reply service is created by the JMS 1.1 Acquire Connection and Session service. The JMS 1.1 Acquire Connection and Session is also used to release to the pool. |
| Application requirements | This depends on how you want to use the JMS. You must create the JMS connection factory and destinations if you want to use JNDI. |
| Initiates business processes? | No |
| Invocation | This service can only be used in a business process. |
| Business process context considerations | None |
| Returned status values | None |

| Restrictions | The JMS 1.1 specification states, a temporary destination is used to receive a message. Typically the JMS 1.1 Request Reply is used to make sure that the reply comes within a specified period. With multiple business processes running, the system might exhaust all the threads allocated for running the business process. You can use the JMS 1.1 Request Reply to specify:<br>• Time-out for the reply<br>• Number for replies expected per request<br>• Permanent destination rather than a temporary destination |
|---|---|
| Persistence level | Since this service operates between the JMS 1.1 Acquire Connection and Session Service and JMS 1.1 Release Connection and Session service, the service has minimum persistence. |
| Testing considerations | None |

## Configuring the JMS 1.1 Request Reply Service

You must specify field settings, using the Admin Console and the GPM.

Use the field definitions in the following table to create a new configuration of the JMS 1.1 Request Reply service, or to set up the configuration provided with the system. Some fields are available in both the Admin Console and in the GPM.

| UI Field | 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:<br>• None – Do not include the configuration in a service group at this time.<br>• 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.)<br>• Select Group – If service groups already exist for this service type, they are displayed in the list. Select a group from the list. |
| | Config | Select the service configuration from the list. |

| UI Field | GPM Field | Description |
|---|---|---|
| | BOUNDED_HASHMAP_SIZE | Number of requests that can be sent from the JMS 1.1 Request Reply service instance without receiving any replies for the requests. After the requests have been sent, the business process that uses this request reply service will not be able to send a message until a reply is received for the pending requests. This maintains an equilibrium between requests and replies. This parameter is only valid if using the request reply with a permanent destination. Default value is 10. |
| | JMSCorrelationID | Unique correlation ID identifies a message. Not required for permanent destination mode, as it will be set automatically. |
| | JMSDeliveryMode | Specifies the JMS Delivery Mode. Valid value is any string value. Optional. |
| | JMSDestination | Specifies the JMS Message Property Destination. Valid value is a destination name. Optional. |
| | JMSExpiration | Specifies the JMS Message Property Expiration. Valid value is a long string value. Optional. |
| | JMSMessageID | Specifies the JMS Message Property Message ID. Valid value any string value. Optional. |
| | JMSPriority | Specifies the JMS Message Property Priority. Optional. Valid value is an integer. Optional. |
| | JMSRedelivered | Specifies the JMS Message Property delivery flag. Optional. Valid value is true or false. |
| | JMSTimestamp | Specifies the JMS Message Property Timestamp. Optional. Valid value is a long string value. |
| | JMSType | Specifies the JMS type. Optional. Valid value is any string value. Some JMS providers use a message repository that contains the definitions of messages sent by applications. The JMSType header field may reference a message's definition in the provider's repository. WebLogic JMS currently does not have a message definition repository. If the message will not be forwarded to a foreign vendor, the JMSType can be used to store an arbitrary value, similar to the usage for JMSCorrelationID. |

| UI Field | GPM Field | Description |
|---|---|---|
| | MSG_TYPE | Type of JMS Message to be sent. Required. Valid values are:<br>• Bytes Message<br>• Object Message<br>• Text Message<br>• Stream Message<br>No default value. |
| | OUTPUT_FILE_NAME | User specified file name for the data that gets received from the destination. If you are receiving documents in batch mode or from the Async Adapter, you can use %^ as the output file name. When using the %^, the output file name is assigned in the format <nodename>_yyyymmddhhmmsslll. |
| | REQUEST_REPLY_EXPECTED_ REPLIES | Number of expected replies for a given request. This is valid only when using permanent destination. For temporary destination the value is always 1. If the number of requested replies specified is not received within the time-out period a request timeout exception is thrown. Default value is 1. |
| | REQUEST_REPLY_RECEIVE | Name of the reply destination. Any valid destination. Required only when using permanent destination mode. |
| | REQUEST_REPLY_SEND | Name of the request destination. Destination is retrieved depending on whether the JMS 1.1 Acquire Connection Session service uses JNDI or Non JNDI. If the request destination is not specified at the JMS 1.1 Acquire Connection Session Service level, then this destination is mandatory. If specified in both places, this parameter takes precedence. |
| | REQUEST_REPLY SEND_TYPE | Destination type of request destination. Required for both permanent and temporary destination mode. |
| | REQUESTER_TYPE | Whether to use a single permanent destination to receive all the replies or a temporary destination per request. Value values are:<br>• Permanent Destination<br>• Temporary Destination (This value is not supported if you are using SonicMQ 7.5 or JBOSS 4.2.2.) |

| UI Field | GPM Field | Description |
|---|---|---|
| | RR_CONC_CONSUMERS | Concurrent number of consumers that can be present on this reply destination at any given time. This is a performance enhancement parameter. Required only when using permanent destination mode. Valid only when using permanent destination. Default value is 1. For permanent destination, the number of concurrent consumers + bounded hash map size should be less than the number of sessions specified in the acquire connection session service. Ideally, the number of sessions is 2*(bounded hash map size + concurrent consumers). With a setting greater than 1, multiple consumers will process replies and the order in which messages where requested may not be the order in which they are responded to. The same applies to the number of responders, if it is greater than 1, the order is not maintained. |
| | TIMEOUT | Time in milliseconds for which the request should wait when expecting a reply. If a reply is not received within this time, a request timeout exception is thrown. Any valid integer. Default is 120000 milliseconds. Required for permanent destination mode. |

## Business Process Example

### Example 1

This business process illustrates sending a request and waiting for the time specified in the TIMEOUT (milliseconds) parameter. If no response is received within the timeout parameter, the business process will fail with a request time out exception. This business process uses a permanent destination rather than a temporary destination to receive a response.

This business process uses the following parameters to specify:
- Timeout for reply
- Number of expected replies
- Permanent destination
- Bounded hash map size to maintain equilibrium between request replies

```
<process name="default">
   <sequence>
      <operation name="JMS11AcquireConnSessionService">
            <participant name="testRegressionsAcquireConnSessionService" />
            <output message="JMS11AcquireConnSessionServiceInputMessage">
                <assign to="." from="*"></assign>
            </output>
            <input message="inmsg">
```

```
                                    <assign to="." from="*"></assign>
                              </input>
                        </operation>
                     <operation name="JMSReqRepService">
                           <participant name="JMSRequestReplyPermanentDestinationService" />
                           <output message="JMSReqRepServiceInputMessage">
                                    <assign to="." from="*"></assign>
                                    <assign to="BOUNDED_HASHMAP_SIZE">10</assign>
                                    <assign to="MSG_TYPE">TextMessage</assign>
                                    <assign to="REQUEST_REPLY_EXPECTED_REPLIES">1</assign>
                                    <assign to="REQUEST_REPLY_RECIEVE">ReplyDestination</assign>
                                    <assign to="REQUEST_REPLY_SEND">RequestDestination</assign>
                                    <assign to="REQUEST_REPLY_SEND_TYPE">QUEUE</assign>
                                    <assign to="REQUESTER_TYPE">Permanent Destination</assign>
                                    <assign to="RR_CONC_CONSUMERS">1</assign>
                                    <assign to="TIMEOUT">30000</assign>
                              </output>
                                <input message="inmsg">
                                    <assign to="." from="*"></assign>
                                </input>
                        </operation>
                     <operation name="JMS11ReleaseConnSessionService">
                              <participant name="testRegressionsReleaseConnSessionService" />
                              <output message="JMS11ReleaseConnSessionServiceInputMessage">
                                    <assign to="." from="*"></assign>
                                    <assign to="RELEASE_ACTION">NONE</assign>
                                </output>
                                <input message="inmsg">
                                      <assign to="." from="*"></assign>
                                    </input>
                        </operation>
                </sequence>
        </process>
```

## Example 2

This business process uses a temporary destination rather than a permanent destination to receive a response.

```
<process name="default">
  <sequence>
    <operation name="JMS11AcquireConnSessionService">
      <participant name="testRegressionsAcquireConnSessionService"/>
      <output message="JMS11AcquireConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMSReqRepService">
      <participant name="JMSRequestReplyTemporaryDestination"/>
      <output message="JMSReqRepServiceInputMessage">
        <assign to="." from="*"></assign>
        <assign to="MSG_TYPE">TextMessage</assign>
        <assign to="REQUEST_REPLY_SEND">RequestDestination</assign>
        <assign to="REQUEST_REPLY_SEND_TYPE">QUEUE</assign>
        <assign to="REQUESTER_TYPE">Temporary Destination</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMS11ReleaseConnSessionService">
      <participant name="testRegressionsReleaseConnSessionService"/>
      <output message="JMS11ReleaseConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
```

```
        <assign to="RELEASE_ACTION">NONE</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

## Frequently Asked Questions

## What is the difference between PERMANENT and TEMPORARY DESTINATION mode?

**TEMPORARY DESTINATION**

If the JMS API specifies that the Topic or Queue Requestor is used for a request reply, the message is sent to the request/send destination specified in the service. A temporary destination is created for each message sent and is set in the JMS Reply To header of the sent message. A third party consumer can receive this message and reply back at the temporary destination specified in the reply to header. As soon as the reply is received on this temporary destination, the JMS 1.1 Request Reply flow is marked as complete.

Temporary destination limitations include:

- For every request sent, only one reply can be received.
- The JMS 1.1 Request Reply service remains in the waiting state unless a reply is not received. The business process remains in active state until a reply is not received. You need to determine if having the business process staying in the active state impacts your system.
- No throttling can be attained between sends and receives. This means, if you have a fast producer (sending messages) and slow consumer (replying to these messages) then you might end up overloading the provider with too many sends.
- This has the overhead of creating a new temp destination for each request which can be resource intensive.

**PERMANENT DESTINATION**

For the permanent destination mode, the customer provides a receive/reply destination along with the send/request destination. A message is sent to the request destination but instead of setting a temporary destination in the reply to header, the reply/receive destination is specified in the header. A unique correlation id is set in the message header to ensure that we are able to match a request to a response. Additionally, an async receiver is started at the reply/receive destination that collects the replies.

Advantages of using PERMANENT DESTINATION mode include:

- More than one reply can be received for a single request.
- Time out can be specified for a request. If no reply is received within the time out, then the business process is marked as failed.
- A parameter called the BOUNDED HASHMAP LIMIT can be used to maintain throttling between send and receive messages. The BOUNDED HASHMAP LIMIT defines the maximum number of messages that can sent to the provider at any given point in time.

### How do I stop the async receiver started in PERMANENT DESTINATION mode by JMS 1.1 Request Reply service?

An async receiver starts listening on the reply destination assigned in the JMS 1.1 Request Reply service as soon as the first request/message is sent. It continues to remain active until manually shut down. The async receiver can also be stopped by using the JMX or using the JMS 1.1 Release Connection and Session service. The pools span the Async Receiver instances, so shutting down a given Async Receiver only releases the resources back to the pool. The pool to a large degree has its own life cycle that is independent of a given adapter. Thus, to shutdown the pool you have to do it explicitly.

The following is an example of the JMS 1.1 Release Connection And Session workflow:

Start > JMS 1.1 Release Connection And Session Service > Stop

In the JMS 1.1 Release Connection and Session service:
- RELEASE MULTIPLEXER is ACTION
- Select the async receiver from the drop down
- Run the business process to shutdown async receiver

The async receiver in the drop down has a format of replyDestination_URL_numberOfConccurentConsumers_ BoundedHashMapSize_uniqueID. There is a chance that customer might have created different receivers to the same destination but with different number of concurrent Consumers and different bounded hash map size.

### When should I stop the async receiver started in PERMANENT DESTINATION mode by JMS 1.1 Request Reply service?

You should manually shut down the async receiver:
- When all the business processes using the JMS 1.1 Request Reply service have finished successfully and the none of these business processes need to be run again.
- If you edited any of the pool parameters of the JMS 1.1 Acquire Connection and Session service that was a part of the business processes that use the JMS 1.1 Request Reply service. Since the async receiver is also setup using the same pool, any changes in this pool lead to the creation of a new async receiver along with the old one. The old async receiver should be shutdown before making any changes to the pool parameters.
- If you edited the BOUNDED HASH MAP LIMIT, REPLY DESTINATION, or the CONCURRENT CONSUMERS parameters in the JMS 1.1 Request Reply service, you need to shut down the async receiver before running a new workflow with the modified parameters. If the async receiver is not shut down before editing any of these parameters, you might end up with more than one async receivers against the same destination and the JMS 1.1 Request Reply service will not work properly. You can confirm this by listing the multiplexers using the JMX. .

### Why am I seeing more than one async receiver in the PERMANENT DESTINATION mode?

If you fail to shutdown the async receiver before changing the BOUNDED HASH MAP LIMIT, REPLY DESTINATION, or the CONCURRENT CONSUMERS

parameters, you will see multiple instances of the async receiver. The request reply might fail as more than one receiver is listening to the reply destination. You must shutdown the multiplexer before editing any of the parameters.

You can use the JMX to list the number of multiplexers and check if more than one multiplexer is listed against the same destination. Additionally, the multiplexers can be listed using the JMS 1.1 Release Connection and Session service by using the drop down of RR_LIST parameter.

## Do I need to specify the correlation id when using JMS 1.1 Request Reply service?

The Correlation Id is not used in the TEMPORARY DESTINATION mode. If you are using the PERMANENT DESTINATION mode, the correlation id is used to uniquely match a request to a reply. You need to have an unique correlation id for every message sent using Request Reply. Unless customers have a unique mechanism to ensure that they will be able to set a unique correlation id for every request this field should be left blank as a unique correlation id is automatically generated in code and set. If the customer provides own correlation id the default one is overridden.

## How is a response to a request sent using JMS 1.1 Request Reply service?

For TEMPORARY DESTINATION, the reply should be sent to the destination specified in the reply to header of the received message.

For PERMANENT DESTINATION, the reply should be sent to the destination specified in the reply to header of the received message. The correlation id to the reply should be set to the same as the correlation id of the received/request message.

## What happens if I get a message after the received time out in the PERMANENT DESTINATION mode?

If you are using the PERMANENT DESTINATION mode, the async receiver remains active, unless it is shutdown manually. If you receive more replies than was specified in the number of replies, or if a reply is received after the time out of a request, the replies are not lost. To ensure that none of these messages are lost, you can set a property called ERROR_QUEUE as a message header in the reply. Specify the destination where you want these replies to be redirected in case this reply is received after a time out or if it is more than the number of replies that were expected.

## What happens when the JMS provider goes down?

If you are using the PERMANENT DESTINATION mode, there is always an async receiver listening against the reply/receive queue waiting for messages. If the JMS provider goes down, this async receiver executes the retry logic to reconnect itself and the JMS provider will come back. The number of retries and the amount of time between each retry is defined in the jms11.properties file.

**What is the difference between disabling the Async Receive Service and stopping the Async Receive/shutting down the pool using the JMX or the Release Connection and Session component?**

Shutting down a given async receiver only releases resources back to the pool. The pool to a large degree has its own life cycle that is independent of a given adapter. Thus to shutdown the pool you have to do it explicitly.

# Chapter 126. JMS 1.1 Send Message Service

The JMS 1.1 Send Message service creates a JMS Message Producer and sends a message to the specified destination. The following table provides an overview of the JMS 1.1 Send Message service:

| System Name | JMS 1.1 Send Message service |
|---|---|
| Graphical Process Modeler (GPM) categories) | All Services |
| Description | The JMS 1.1 Send Message service creates a JMS Message Producer and sends a message to the specified destination. The same service is used for both queue and topic. |
| Business usage | The JMS 1.1 Send Message service creates a JMS Message Producer and sends a message to the specified destination. The same service is used for both queue and topic. |
| Usage example | Use this service to send messages to remote Queues or Topics. |
| Preconfigured? | Yes |
| Requires third party files? | The JMS provider jar must be provided to the system, using one of the following methods:<br><br>• Install using the install3rdParty script<br>• Load using the custom classloader<br><br>For JBoss and WebLogic, you must use the custom classloader as these vendor jars contain classes that already exist in system classloader.<br><br>Before you install the jar file, you need to know:<br><br>• Remote JMS server context factory<br>• URL (IP address and port)<br>• Connection factory name<br>• User name and password (if applicable)<br>• Public key for the CA or trusted certificates (if used) from the trading partner for access to their JMS server or system certificate (if used)<br><br>For example, if you need to communicate to an external WebLogic JMS server, you need the jar file that includes the weblogic.jndi.WLInitialContextFactory class. You would obtain the jar file from the corresponding vendor or your trading partner. |
| Platform availability | All supported platforms |
| Related services | This service can be used in conjunction with the JMS 1.1 services that are included in business processes:<br><br>• JMS 1.1 Acquire Connection and Session service and JMS 1.1 Release Connection and Session service are used to start and stop a JMS session within a business process. They must be used as a pair in a business process.<br>• JMS 1.1 Receive Message service is used to get messages from a JMS topic or queue. |

| | | |
|---|---|---|
| Application requirements | None | |
| Initiates business processes? | No. This service cannot be used outside of a business process. | |
| Invocation | None | |
| Business process context considerations | None | |
| Returned status values | None | |
| Restrictions | None | |
| Persistence level | Minimum | |
| Testing considerations | Enable JMS logging from the Admin Console by selecting Operations > System > Logs. | |

## Configuring the JMS 1.1 Send Message Service

You need to create a configuration in the User Interface, and then you must specify field settings in the GPM:

| UI Field | 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:<br>• None - Do not include the configuration in a service group at this time.<br>• 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.)<br>• Select Group - If service groups already exist for this service type, they are displayed in the list. Select the group from the list. |
| | Config | Select the name of the service configuration from the list. |
| | BATCH_ SEND_FILTER | Indicates batch sending mode (sending multiple messages). Optional. Valid value is any string value. |
| | DESTINATION_NAME_WFC | Specifies the Destination Name. If Destination Name is specified in JMS 1.1 Acquire Connection and Session Service, that value overrides the JMS 1.1 Send Message Service Destination Name. Optional. Valid value is any string value. |

| UI Field | GPM Field | Description |
|---|---|---|
| | DESTINATION_TYPE_WFC | Specifies the Destination Type (QUEUE/TOPIC) in the JMS 1.1 Send Service. If Destination Type is specified in JMS 1.1 Acquire Connection and Session Service, that value overrides the JMS 1.1 Send Message Service Destination Type. Optional. Valid value is any string value. |
| | JMSCorrelationID | Specifies the JMS Message Property Correlation ID. Optional. Valid value is any string value. |
| | JMSDeliveryMode | Specifies the delivery mode. Optional. Valid values are:<br><br>• NON PERSISTENT<br>• PERSISTENT |
| | JMSDestination | Specifies the JMS Message Property Destination. Optional. Valid value is a destination name. |
| | JMSExpiration | Specifies the JMS Message Property Expiration. While sending the JMS message, you can use the value specified in this field and set it equal to time to live. The send message remains available in the destination until time to live expires. Optional. Valid value is a long string value. |
| | JMSMessageID | Specifies the JMS Message Property Message ID. Optional. Valid value is any string value. |
| | JMSPriority | Specifies the JMS Message Property priority. Optional. Valid values are 0 to 9. |
| | JMSRedelivered | Specifies the JMS Message Property redelivery flag. Optional. Valid value is true or false. |
| | JMSTimestamp | Specifies the JMS Message Property timestamp. Optional. Valid value is a long string value. |
| | JMSType | Specifies the JMS type. Optional. Valid value is any string value. Some JMS providers use a message repository that contains the definitions of messages sent by applications. The JMSType header field may reference a message's definition in the provider's repository. WebLogic JMS currently does not have a message definition repository. If the message will not be forwarded to a foreign vendor, the JMSType can be used to store an arbitrary value, similar to the usage for JMSCorrelationID. |

| UI Field | GPM Field | Description |
|---|---|---|
| | MSG_TYPE | Describes the payload type. Required. Valid values are: <br>• Bytes Message <br>• Object Message <br>• Text Message <br>• Stream Message <br><br>No default value. |

## Parameters Passed From Business Process to Service

The following parameters (GPM field names) are passed from the business process to the JMS 1.1 Send Message service:

• DESTINATION_NAME_WFC
• DESTINATION_TYPE_WFC
• MSG_TYPE
• BATCH_SEND_FILTER
• JMSCorrelationID
• JMSDestination
• JMSExpiration
• JMSMessageID
• JMSPriority
• JMSRedelivered
• JMSTimestamp
• JMSType
• JMSDeliveryMode

## Business Process Examples

### Example 1

The following example does not override any parameters in the Send Service and only sets the mandatory parameters. It uses JMS11AcquireConnectionAndSessionForQueue service which is UI configured and preset with all the parameters.

```
<process name="jms11SendToQueueRegression">
  <sequence>
    <operation name="JMS11AcquireConnSessionService">
      <participant name="JMS11AcquireConnectionAndSessionForQueue"/>
      <output message="JMS11AcquireConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMS11SendMessageService">
      <participant name="JMS11SendMsgRegression"/>
      <output message="JMS11SendMessageServiceInputMessage">
        <assign to="." from="*"></assign>
        <assign to="MSG_TYPE">TextMessage</assign>
      </output>
      <input message="inmsg">
```

```
          <assign to="." from="*"></assign>
        </input>
      </operation>
      <operation name="JMS11ReleaseConnSessionService">
        <participant name="JMS11ReleaseConnectionAndSession"/>
        <output message="JMS11ReleaseConnSessionServiceInputMessage">
          <assign to="." from="*"></assign>
        </output>
        <input message="inmsg">
          <assign to="." from="*"></assign>
        </input>
      </operation>
    </sequence>
</process>
```

## Example 2

The following example shows how to use the Send service by specifying the
Destination Name and Type and also setting JMS Headers. When this business
process is executed , the message is sent to "SOMEQUEUENAME" with
Correlation ID as "testCorrID" irrespective of what the Destination Name and Type
were set in the Acquire Connection Service.

```
<process name="jms11SendToQueueRegression">
  <sequence>
    <operation name="JMS11AcquireConnSessionService">
      <participant name="JMS11AcquireConnectionAndSessionForQueue"/>
      <output message="JMS11AcquireConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMS11SendMessageService">
      <participant name="JMS11SendMsgRegression"/>
      <output message="JMS11SendMessageServiceInputMessage">
        <assign to="." from="*"></assign>
       <assign to="DESTINATION_NAME">SOMEQUEUENAME</assign>
       <assign to="DESTINATION_TYPE">QUEUE</assign>
       <assign to="JMSCorrelationID">testCorrID</assign>
 <assign to="MSG_TYPE">TextMessage</assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMS11ReleaseConnSessionService">
      <participant name="JMS11ReleaseConnectionAndSession"/>
      <output message="JMS11ReleaseConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

## Example 3

The following example demonstrates batch sending mode. The File System adapter
is used to collect multiple documents which are sent via batch sending mode.

```
<process name="jms11BatchSend">
  <sequence>
    <operation name="JMS11AcquireConnSessionService">
      <participant name="JMS11AcquireConnectionAndSessionForQueue"/>
      <output message="JMS11AcquireConnSessionServiceInputMessage">
      <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
 <operation name="BatchCollect">
      <participant name="JMS11FSA"/>
      <output message="xout">
        <assign to="Action">FS_COLLECT</assign>
        <assign to="collectionFolder">path to collect the files </assign>
        <assign to="bootstrap">false</assign>
        <assign to="collectMultiple">true</assign>
        <assign to="fileModTimeThreshold">0</assign>
        <assign to="deleteAfterCollect">false</assign>
        <assign to="fsa_debug">true</assign>
      </output>
      <input message="xin">
        <assign to="." from="*"/>
      </input>
    </operation>
    <operation name="JMS11SendMessageService">
      <participant name="JMS11SendMsg"/>
     <output message="toService">
     <assign to="." from="*"/>
     <assign to="BATCH_SEND_FILTER">FSA*</assign>
        <assign to="MSG_TYPE">TextMessage</assign>
      </output>
      <input message="fromService">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation name="JMS11ReleaseConnSessionService">
      <participant name="JMS11ReleaseConnectionAndSession"/>
      <output message="JMS11ReleaseConnSessionServiceInputMessage">
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

## Example 4

The following example shows how to send a message using transactions and
rollback if the transaction fails. You can also use other services that can use
transactions like the Lightweight JDBC in combination with the below and make
them commit or rollback together. For example in the below mentioned workflow,
LightweightJDBC service instance can be added before or after JMS 1.1 Send
Message service to ensure that if both sending of message and successful execution
of database query happens then only the transaction gets committed else the
transaction is rolled back.

```
<process name="default">
    <sequence>
            <operation name="Begin Transaction Service">
                <participant name="BeginTransactionService" />
                <output message="BeginTransactionServiceTypeInputMessage">
                        <assign to="." from="*"></assign>
```

```
                    <assign to="START_TRANSACTION">TRUE</assign>
                    <assign to="DISTRIBUTED">TRUE</assign>
                    <assign to="ON_FAULT">ROLLBACK</assign>
            </output>
            <input message="inmsg">
                    <assign to="." from="*"></assign>
             </input>
        </operation>
        <operation name="JMS 1.1 Acquire Connection and Session Service">
                <participant name="JMS11AcquireConnectionAndSessionFor
                        QueueRegression" />
                <output message="JMS11AcquireConnSessionServiceInputMessage">
                        <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
                        <assign to="." from="*"></assign>
                 </input>
        </operation>
        <operation name="JMS 1.1 Send Message Service">
                <participant name="JMS1.1SendMessageService " />
                <output message="JMS11SendMessageServiceInputMessage">
                        <assign to="MSG_TYPE">TextMessag</assign>e
                                <assign to="MSG_TYPE">TextMessage</assign>
                        <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
                        <assign to="." from="*"></assign>
                </input>
        </operation>
        <operation name="End Transaction Service">
                <participant name="EndTransactionService" />
                <output message="EndTransactionServiceTypeInputMessage">
                        <assign to="." from="*"></assign>
                        <assign to="ROLLBACK_TRANSACTION">FALSE</assign>
                        <assign to="END_TRANSACTION">TRUE</assign>
                </output>
                <input message="inmsg">
                        <assign to="." from="*"></assign>
                </input>
        </operation>
        <operation name="JMS 1.1 Release Connection and Session Service">
                <participant name="JMS11ReleaseConnectionAndSession
                         Regression" />
                <output message="JMS11ReleaseConnSessionServiceInputMessage">
                        <assign to="." from="*"></assign>
                </output>
                <input message="inmsg">
                        <assign to="." from="*"></assign>
                </input>
        </operation>
    </sequence>
</process>
```

If you need more information on the Begin and End JTA transaction services, see the Begin and End JTA transaction services documentation.

## Frequently Asked Questions

### How can I set reply to header as temporary destination?

To specify a temporary destination use temporaryQueue// or temporaryTopic// in the reply to field of the JMS 1.1 Send Message Service rather than specifying a destination name.

## How do I set custom JMS Message properties?

In addition to the message headers provided in the JMS 1.1 Send Message service, several other properties can be added to the message header by using assigns. By using the assigns feature, these headers can be used with message selectors to filter what messages to receive, what messages to ignore, and can be used for routing in many Enterprise Application Integration (EAI) patterns. If the message selector is blank, then all of the messages are candidates to receive irrespective of any header value.

Example: Consider a bookstore that sends messages to a queue whenever it runs out of inventory. There are several vendors that are listening on this queue which are capable of supplying this inventory to this bookstore. The criterion on which a particular vendor is selected depends on the amount of order (order quantity needed).

This can be achieved easily by sending a message with a custom header specifying the minimum quantity, and each of the vendors can have a message selector that reads this header and depending on this header picks up the message.

## What are the various levels where I can set destination override?

There are three levels of destination overrides. You can specify the destination to send the message to:
- JMS 1.1 Acquire Connection and Session service level
- JMS 1.1 Send Message Service level
- Both the JMS 1.1 Acquire Connection and Session service level and the JMS 1.1 Send Message Service level. When both are specified, the JMS 1.1 Send Message Service level destination takes precedence. This provides the ability to make where you send a message dynamic (a variable that can be passed into the business process versus being hardcoded).

The advantage of specifying at the send level is that same acquire service can be reused in multiple work flows. This is useful when send message service is running in a loop.

In addition, if the destination is not specified at both these levels but if there is a JMS 1.1 Receive Message service in the work flow that receives a message with a Reply To header set in it, the JMS 1.1 Send Message Service can send the message to the destination in the Reply To header. This is given the last precedence. The following is an example workflow:

Start > JMS 1.1 Acquire Connection And Session Service > JMS 1.1 Receive Message Service (receiving a message from some destination with message having reply to header set on it) > JMS 1.1 Send Message Service (with destination not specified at either acquire or send level but will pick the destination in the reply to header of the received message) > JMS 1.1 Release Connection And Session Service

## What are the various modes to send bulk messages?

You can send bulk messages using the batch mode or you can loop the JMS 1.1 Send Message Service. Sending messages in batch mode is not a part of JMS 1.1 specification, but is provided by Sterling Integrator to enable customers to send multiple messages simultaneously.

For example, you may need to send all the documents in a folder as messages. You can accomplished this by either picking one document at a time, sending it as a message or by using the Batch mode. In Batch mode, you can pick all the messages simultaneously and then send them. Batch mode prevents the overhead of creating a new producer, new session, and returning them for each message as the same producer and session can be used to send all the messages.

In addition, when using Batch mode with transactions, the session only has to be committed once. This is similar to running multiple database queries and calling the database commit only once.

Batch mode provides better performance results as compared to single message mode.

## What happens when the JMS provider goes down?

As the JMS 1.1 Send Message service is preceded by JMS 1.1 Acquire Connection and Session service that carries out the work to create a connection and session, if the provider is down, the work flow fails with an unable to set up the connection exception.

## What happens when a business process fails with an exception (say JMS or some other exception) before Release connection and Session component is called, will our JMS adapter release the connections automatically or do you need to code the exception block in business process to release the connections manually during a exception scenario?

Even in this scenario, connections and sessions are returned back to the pool.

## How do I use transactions in JMS 1.1 Send Service?

Transactions can be used when sending messages, as they ensure high throughput when sending bulk messages. You will need to call the End Transaction service before JMS 1.1 Release Connection and Session service.

## How can I send messages to temporary destination and receive it?

To send a message to a temporary queue, select the destination type as temporary queue. To receive a message sent to a temporary destination by send message service, the JMS 1.1 Receive Message service should be used in this order:

Start > JMS 1.1 Acquire Connection And Session Service > JMS 1.1 Send Message Service (with destination type as temporary queue or topic and no destination name specified) > JMS 1.1 Receive Message Service (with destination type as temporary queue or topic and no destination name specified and it should be placed between the same acquire and release service) > JMS 1.1 Release Connection And Session Service

This workflow is required as the JMS 1.1 specification provides that a message can be received from a temporary destination by the same connection that sends the message.

# Chapter 127. Lightweight Directory Access Protocol (LDAP) Adapter

The following table provides an overview of the LDAP adapter:

| System name | None |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | Enables Sterling Integrator to communicate with local or remote LDAP servers using a Java Naming Directory Interface (JNDI). You can operate on data entries contained on an LDAP server, but you cannot change the underlying structure of those entries. The LDAP adapter supports Create, Read, Update, and Delete (CRUD) operations. |
| Preconfigured? | No |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | No |
| Application requirements | No |
| Initiates business processes? | No |
| Invocation | Runs as part of a business process. |
| Restrictions | The LDAP adapter supports LDAP versions 2 and 3 to the extent that the Sun LDAP/JNDI libraries do, except as noted in the following points:<br>• The following standard LDAP operations are not supported:<br>  - Adding new entry types<br>  - Adding or removing attributes from an entry<br>  - Referrals<br>• Multi-value fields are supported for read, create, and update operations, but all values are treated as a single replacement value in the update operation; that is, the LDAP adapter cannot support updates of just one value in a multi-value field.<br>• The LDAP adapter supports only the simple authentication type.<br>• To prevent usernames and passwords from being accessible to other system users, enter usernames and passwords as instance variables only, not the BPML itself. |

## Requirements

To use the LDAP adapter, you must meet the following requirements:

## Knowledge Requirements

To set up and use the LDAP adapter, you should know how to:
• Use the LDAP data model

- Use the Map Editor and the Translation service
- Apply XML concepts

## Supported LDAP Versions

The LDAP adapter supports LDAP versions 2 and 3 to the extent that the Sun LDAP/JNDI libraries do, except as noted in the following points:
- The following standard LDAP operations are not supported:
  - Adding new entry types
  - Adding or removing attributes from an entry
  - Referrals
- Multi-value fields are supported for read, create, and update operations, but all values are treated as a single replacement value in the update operation; that is, the LDAP adapter cannot support updates of just one value in a multi-value field.
- The LDAP adapter supports only the simple authentication type.

## System Requirements

For the LDAP adapter to work correctly, verify that:
- You have a valid logon ID and password and can access the remote LDAP server
- You can make a physical connection to the LDAP server from Sterling Integrator at run time

## How the LDAP Adapter Works

Using LDAP directories is a popular method for storing and retrieving simple data in a hierarchical structure. LDAP works well with data on a wide area network (WAN).

**Note:** The LDAP adapter is not used in the authentication of external users of Sterling Integrator. This process does use an LDAP server, but not the LDAP adapter.

## Java Naming Directory Interface (JNDI)

LDAP servers organize data into a hierarchical structure. An LDAP directory enables you to search a structured data repository and is optimized for read operations, unlike databases. Each record in the informational hierarchy can contain one or more fields or attributes. Each attribute can contain one or more values.

LDAP servers are not databases, although they can use databases to implement data storage. This distinction is important because LDAP may not support many sophisticated database features, such as advanced relational queries with table joins and transactional integrity across multiple operations. At present, the adapter accesses data in an LDAP server through the JNDI/LDAP API. The JNDI/LDAP API enables selection of LDAP data elements by name.

In general terms, LDAP is an example of a schema-based Operational Support System (OSS) as opposed to a service-based OSS. To make LDAP more service-based, Sterling Integrator overlays a standard service layer named CRUD (Create, Read, Update, and Delete) to manipulate data.

The service layer works as follows:

- Create – Adds a new entry to a directory and provides data for any attribute that already exists in the entry.
- Read – Provides an entry search filter; the retrieved data is in Directory Service Markup Language (DSML) format.
- Update – Modifies an LDAP entry. You need to provide a base distinguished name (baseDN) to identify the entry, and the names and values of the attributes to update.
- Delete – Deletes an entry from the LDAP server.

## Accessing Data

To access data, the LDAP adapter needs these items:

- Service to perform – Create, Read, Update, Delete
- A unique name that specifies a record on the LDAP server
- Field names within the record

## Adapter Process

The following steps summarize how the LDAP adapter works:

1. The LDAP adapter sends a request to the LDAP server.
2. The adapter takes the results returned from the LDAP server and places them back into the business process context.
3. The adapter passes the updated internal business process context back to the business process.
4. The adapter is ready to process the next request.

The following figure shows how the LDAP adapter communicates with the LDAP server within a business process:
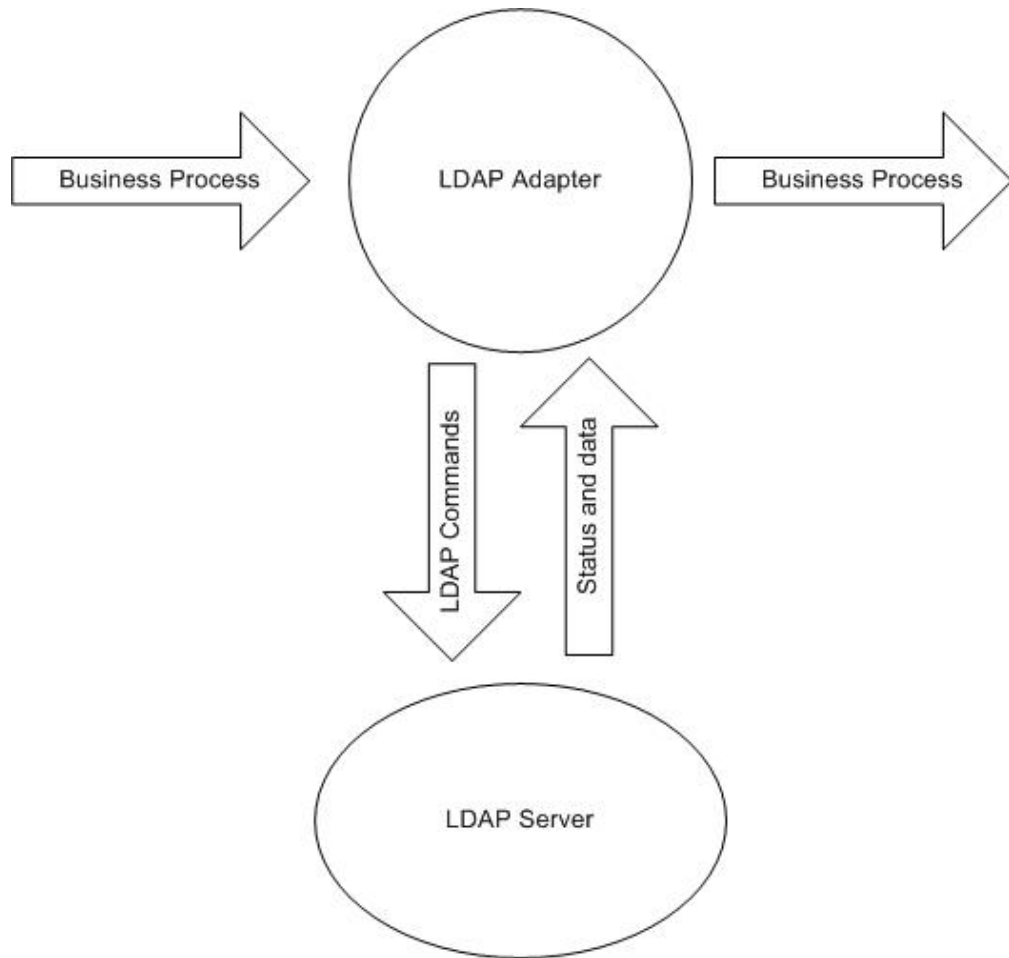
The following steps summarize how the LDAP adapter communicates with the LDAP server within a business process:

1. The Translation service checks the translation object (.txo) in to Sterling Integrator for later use by the Translation service.
2. When initiating the business process, the user supplies the name and location of the customer input document that has the necessary information to be retrieved, such as the customer name and the name of fields.
3. From the business process, the business process engine (BPE) receives the name of the Translation object (identified from the list of maps checked in to Sterling Integrator).
4. When the business process is started, the BPE starts the Translation service. The Translation service builds the XML file for the LDAP adapter with the necessary information from the customer document.
5. The LDAP adapter uses the information in the file to make a connection with the LDAP server and retrieve the required information by making appropriate calls.
6. The LDAP adapter constructs an XML document with the retrieved data.
7. The XML document is passed to the business process.
8. Sterling Integrator performs the next step in the business process.

For example, consider the following scenario. You have customer information stored in an LDAP server (which has an internal database). To provide the sales

department with the customer contact information from the LDAP database, use the LDAP adapter to access this information and then write the information to disk using the File System adapter.

## Implementing the LDAP Adapter

To implement the LDAP adapter:

1. Create an LDAP adapter service configuration. For information, see *Managing Services and Adapters*.
2. Configure the LDAP adapter. For information, see *Configuring the LDAP Adapter*.
3. Create XML documents, as necessary. For information, see *Creating XML Documents for the LDAP Adapter*.
4. Use the LDAP adapter in a business process.

## Configuring the LDAP Adapter

The following tables describe fields used to configure the LDAP adapter in 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. |
| Host Name (hostName) | Name or IP address of the host running the LDAP server. Can be overridden in a business process or process data. |
| Port (port) | IP port number on the host. Default is 389. Can be overridden in a business process or process data. |
| Read Timeout in Secs (readTimeOut) | Timeout value in seconds. Default is No limit (0). |
| Max Number of Records to Read (maxReadRecords) | Maximum number of records to return from the LDAP server. Zero (0) means no limit is applied. |
| Set Authentication? (LDAPAuthentication) | Whether you will attempt to connect to the LDAP server with authentication, or anonymously. |
| Login Name (loginName) | Login name for the host LDAP server. Can be overridden in a business process or process data. |
| Password (password) | LDAP server password for the associated login name. Can be overridden in a business process or process data. |

## Creating XML Documents for the LDAP Adapter

For LDAP adapter business processes, the XML document passed into the adapter determines the operation to start on the LDAP server. The XML document must correspond to one of four Document Type Definitions (DTDs), which define the CRUD operations (Create, Read, Update, or Delete). In other words, the DTDs verify that the XML is correct for the operation it is written for.

The LDAP adapter provides the DTDs that define the XML that is passed to and received from the adapter. The Map Editor and the Translation service use these DTDs to verify data conversions to and from customer formats.

To check out the DTDs from Sterling Integrator and load them to your local disk:

1. From the **Deployment** menu, select **Schemas**.
2. In the XML Schemas window, search for the LDAP DTDs.

   They are named LDAPCreate.dtd, LDAPRead.dtd, LDAPUpdate.dtd, LDAPDelete.dtd, and dsml.dtd.
3. Click **Source Manager**.
4. Save each DTD to your local disk.

## XML Construction

When constructing XML for the LDAP adapter, remember the following points:

- In the request element, the operation attribute specifies the operation to be performed (Create, Read, Update, or Delete), and:
  - The Base Distinguished Name (baseDN) attribute is different for each operation.
  - The Scope attribute specifies search scope and is used only in Read operations.
- Parameter elements identify fields, and:
  - Name attributes identify the field name.
  - Usage attributes specify whether the data is input, output, or search.
  - Type attributes specify the type of data to be sent in the output parameters (for example, text/none, bin/base64).

    **Note:** Usage attributes other than those specified for a particular operation are discarded. For example, if a request XML for a Create operation has input and output usage attributes, the input attribute is discarded.

## Examples

The following XML excerpts are examples of input documents for Create, Read, Update, and Delete operations.

**Create Operations**

The baseDN attribute identifies the record to be created. The usage attribute is always `output` for Create operations.

Two values exist for Type attributes in Create operations:
- text/none – Intended for ordinary textual data with no encoding. Default.
- bin/base64 – For binary data. Encode as `base64` in the content of the param tag.

The following example shows a Create operation:

```
<LDAPAdapter>
  <request operation="Create" baseDN="uid=jblow, ou=People, o=isg.stercomm.com">
    <param.1 name="objectclass" type="text/none" usage="Output">top</param.1>
    <param.2 name="objectclass" type="text/none" usage="Output">
        person
    </param.2>
    <param.3 name="ou" type="text/none" usage="Output">People</param.3>
```

```
    <param.4 name="mail" type="text/none" usage="Output">jb@ab.com</param.4>
    <param.5 name="uid" type="text/none" usage="Output">jblow</param.5>
    <param.6 name="sn" type="text/none" usage="Output">Blow</param.6>
    <param.7 name="givename" type="text/none" usage="Output">Joe</param.7>
  </request>
</LDAPAdapter>
```

### Read Operations

The baseDN attribute identifies the highest point in the hierarchy to begin the search, and the Scope attribute defines the extent of the search. The Scope attribute values are:

- subTree
- base
- oneLevel

The search filter is a query string and is denoted with a param element where the Usage attribute is `search`. Use the following symbols to define the search:

| Symbol | Represents |
|---|---|
| Parentheses ( ) | Enclosed group of compares |
| Ampersand & | Logical operator AND |
| Pipe \| | Logical operator OR |
| Exclamation point ! | Logical operator NOT |

The logical operator should appear before the parentheses enclosing the group of compares that the logical operator affects. For example:

```
(&(cn=X)(sn=Y))
```

This example means that cn is equal to X and sn is equal to Y.

One param element must have a usage attribute value of `search` and must contain the search specification as defined by the LDAP model. The rest of the param elements, if any, specify the field names to be retrieved from the LDAP server for the records that match the search filter.

The following example shows a retrieval for the cn field and the jpegphoto field:

```
<LDAP Adapter>
 <request scope="subtree" operation="Read" baseDN="uid=jblow,
ou=People, o=isg.stercomm.com">
  <param.1 usage="Search">(&(objectclass=person)(sn=Blow))</param.1>
  <param.2 name="jpegphoto" usage="Input"/>
  <param.3 name="cn" usage="Input"/>
 </request>
</LDAP Adapter>
```

### Read Output Documents

Because Read is the only operation that has data returned, an output document is associated with this operation. The output document is written in DSML and is added to the business process context when the operation successfully completes. DSML is a standard representation of directory information in XML format. The LDAP adapter is compatible only with DSML Version 1.0.

DSML is intended to be a simple XML schema definition that enables directories to publish basic profile information. Find the full specifications for DSML at www.dsml.org.

The following example shows DSML for a Read output document:

```
<dsml>
 <directory-entries>
  <entry dn="uid=scarte2, ou=People, o=isg.stercomm.com">
   <attr name="telephonenumber">
    <value>+1 408 555 6022</value>
   </attr>
   <attr name="mail">
    <value>scarte2@isg.stercomm.com</value>
   </attr>
   <attr name="uid">
    <value>scarte2</value>
   </attr>
  </entry>
 </directory-entries>
</dsml>
```

### Update Operations

The baseDN attribute identifies the record to be updated. Two values exist for Type attributes in Update operations:

- text/none – Intended for ordinary textual data with no encoding. Default.
- bin/base64 – Intended for binary data. Encode as base64 in the content of the param element.

The parameter element usage attribute value is always output.

For multivalue fields, multiple param elements can have the same name, but the content is different for each. The following example shows a multivalue **first name** field:

```
<LDAPAdapter>
 <request operation="update" baseDN="uid=jblow, ou=People,
o=isg.stercomm.com">
  <param.1 name="employeenumber" type="text/none" usage="Output">
    1234
    </param.1>
  <param.2 name="firstname" type="text/none" usage="Output">Joe</param.2>
  <param.3 name="firstname" type="text/none" usage="Output">Joseph</param.3>
  <param.4 name="firstname" type="text/none" usage="Output">Joey</param.4>
 </request>
</LDAPAdapter>
```

### Delete Operations

The baseDN attribute identifies the record to be deleted. The following example shows a Delete operation:

```
<LDAPAdapter>
 <request operation="Delete" baseDN="uid=jblow, ou=People,
o=isg.stercomm.com">
 </request>
</LDAPAdapter>
```

# Chapter 128. Lightweight Java Database Connectivity (JDBC) Adapter

The following table provides a high-level overview of the Lightweight JDBC adapter:

| System name | None |
|---|---|
| Graphical Process Modeler (GPM) category | All services, Translation |
| Description | Enables you to retrieve data from a JDBC-compliant database or update a JDBC-compliant database as part of a business process within the application, using one of the following methods: |
| | • Specify XPath queries to retrieve information from process data and set the Lightweight JDBC (LWJDBC) adapter parameters in a business process. |
| | Use XPath queries to set the Lightweight JDBC adapter parameters in a business process, allowing for reuse of the same instance of a Lightweight JDBC adapter configuration multiple times throughout a business process. To use XPath queries, you need to first load into process data the data which you want to use to set the Lightweight JDBC adapter parameters. You can use either the XML Encoder service or DocToDOM function in your business process to load into process data a primary document containing the data you want to use for XPath queries. |
| | • Specify constants in the Lightweight JDBC adapter configuration. |
| | Unlike the JDBC adapter, the Lightweight JDBC adapter does not use a map in which the SQL statements are fixed. The Lightweight JDBC adapter enables you to specify parameters in the SQL statement that are supplied at run time. For information about the JDBC adapter, see *Java Database Connectivity (JDBC) Adapter*. The Lightweight JDBC adapter executes a query, and you receive an XML document that contains the results. This document becomes the primary document in the business process. Then you can either: |
| | • Load the XML document into process data. |
| | • Carry the XML document forward into another service in the business process. |
| Business usage | Use the Lightweight JDBC adapter if you want to retrieve or manipulate data and need the flexibility of specifying various SQL queries. Use the JDBC adapter if you have complex output and you want to manipulate the data from a database and control the structure of the output. |
| Preconfigured? | No |
| Requires third party files? | You must configure a connection to an external database for the Lightweight JDBC adapter. |
| Platform availability | All supported application platforms |
| Related services | None |

| Initiates business processes? | Yes |
|---|---|
| Invocation | Runs by a schedule or business process |

### How the Lightweight JDBC Adapter Works

The Lightweight JDBC adapter can start a business process or it can be used in a business process. This service can be scheduled to run at weekly or timed intervals.

The following figure shows how the Lightweight JDBC adapter communicates with a JDBC-compliant database:
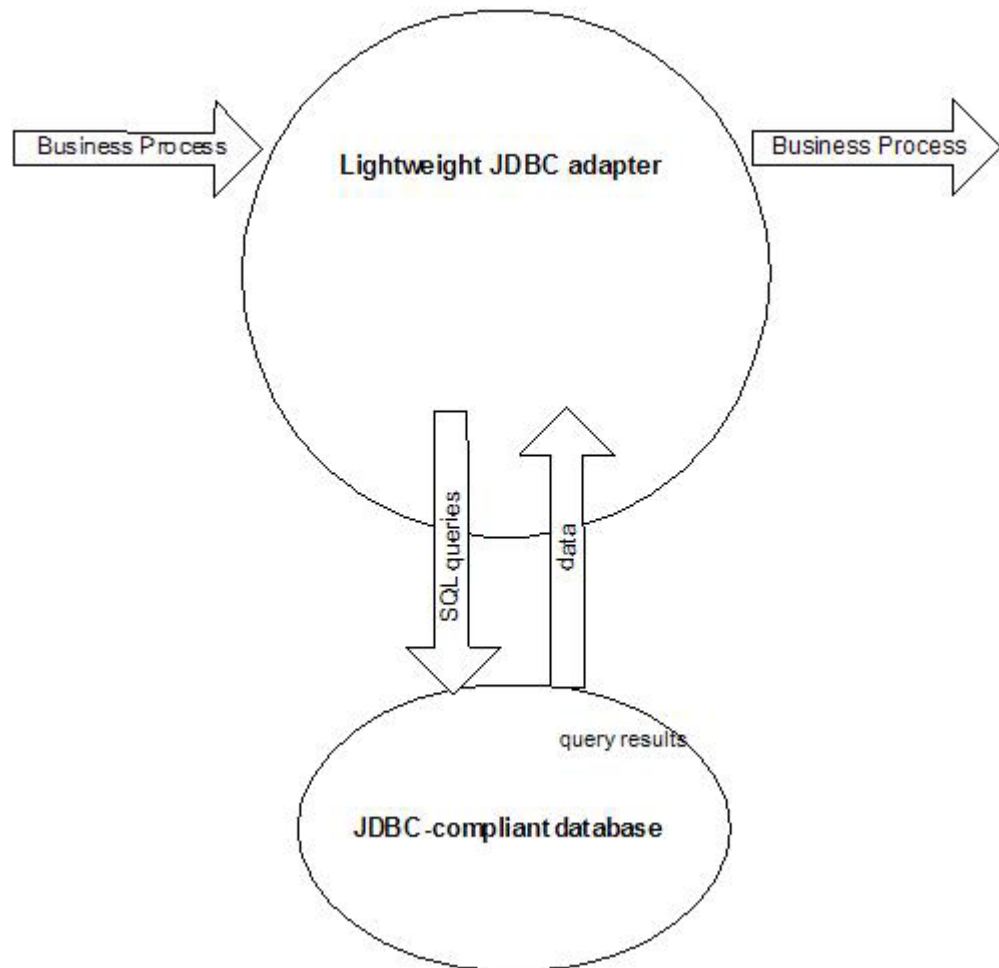


### Business Scenario

Your company receives information from an external database about a customer that needs to be either added (if the customer does not exist) or updated in your internal customer database. The customer information arrives in XML format.

### Business Solution Example

The following approach is used to solve the business scenario:

1. Configure a Lightweight JDBC adapter instance to be included in a business process. This one instance of the Lightweight JDBC adapter is used multiple times throughout the example business process to perform the following tasks:
   - Query the database for the customer ID passed into the business process to determine if the customer currently exists in the internal customer database.
   - Add the customer information to the customer database if the customer does not exist.
   - Update the customer information in the customer database if the customer does exist.
2. Create a business process that determines whether to add or update the customer information to the internal customer database. The Lightweight JDBC adapter instance configured above is used in this business process to perform the query, add, and/or update.
3. Configure a second Lightweight JDBC adapter instance to retrieve customer records from the external database and to start the business process created in the previous step. This Lightweight JDBC adapter instance is also scheduled to run at 10 p.m. daily to retrieve customer records from the external database and return the results to the business process.

## GPM Example

The following example shows a solution to the business scenario using the GPM.

A Lightweight JDBC adapter has been configured and scheduled to retrieve records from an external database. The result from the query is the primary document to this business process.
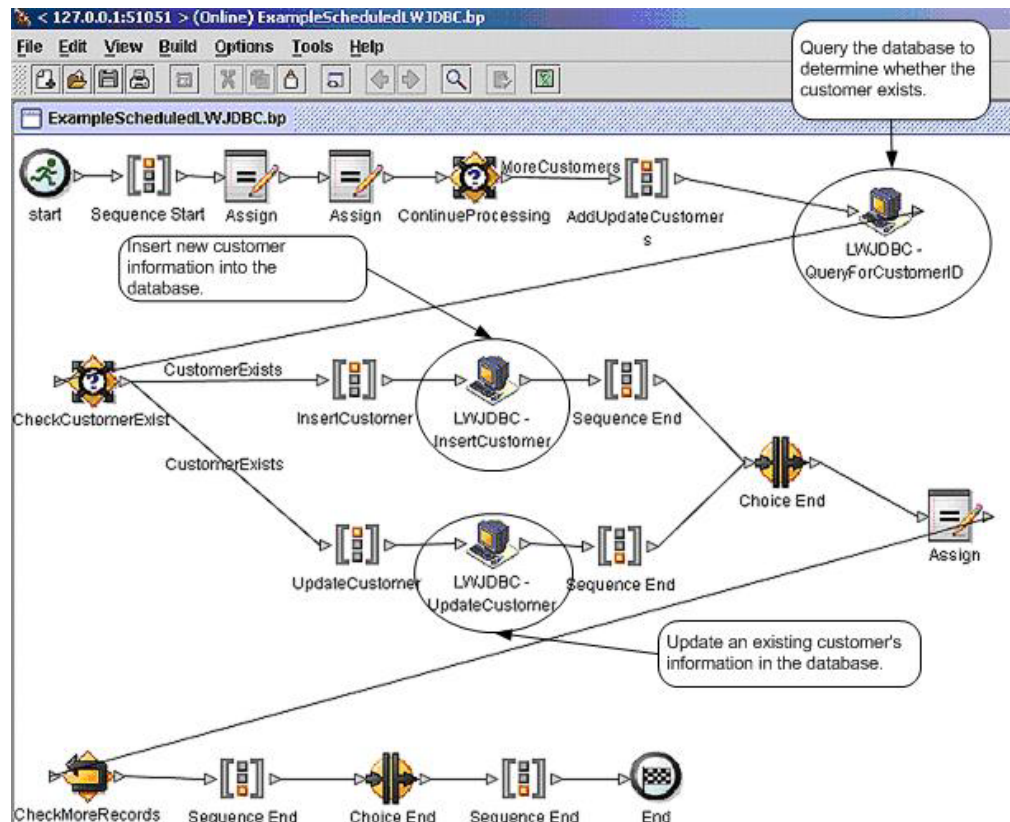
The Assign statements are used to retrieve the customer information from the primary document and place the information into process data. The parameters specified for the Lightweight JDBC adapter, which includes customer information retrieved from process data and SQL statements, are input to the adapters. A primary document containing the results from the SQL statements issued against the database is output from the Lightweight JDBC adapter instances and is passed to the next step in the business process.

For information about the Lightweight JDBC parameters, see *Implementing the Lightweight JDBC Adapter*.

## Business Process Modeling Language (BPML) Example

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

```
<process name="ExampleScheduledLWJDBC">
<rule name="MoreCustomers">
        <condition>NumberOfRecords &gt; 0</condition>
</rule>
<rule name="CustomerExists">
        <condition>number(/ProcessData/NumberOfCustomers) = 0 </condition>
</rule>
<sequence>
        <assign to="NumberOfRecords" from="number(count(DocToDOM(PrimaryDocument)/Customer/
customer_id))" append="true"/>
        <assign to="CustomerInformation" from="DocToDOM(PrimaryDocument)" append="true"/>
        <choice name="ContinueProcessing">
        <select>
                <case ref="MoreCustomers" activity="AddUpdateCustomers"/>
        </select>
        <sequence name="AddUpdateCustomers">
        <operation name="LWJDBC - QueryForCustomerID">
        <participant name="ExampleLWJDBCBusinessProcess"/>
        <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="number(/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_id/text())"/>
        <assign to="paramtype1">Integer</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="result_name">RootResultXmlElement</assign>
        <assign to="row_name">ResultsFromSQL</assign>
        <assign to="sql">SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2 WHERE CUSTOMER_ID = ?</
assign>
        <assign to="." from="*"/>
    </output>
        <input message="inmsg">
        <assign to="NumberOfCustomers" from="DocToDOM(PrimaryDocument)/ResultsFromSQL/
CUSTOMERS" append="true"/>
        </input>
        </operation>
        <choice name="CheckCustomerExist">
        <select>
                <case ref="CustomerExists" activity="InsertCustomer"/>
                <case ref="CustomerExists" negative="true" activity="UpdateCustomer"/>
        </select>
        <sequence name="InsertCustomer">
        <operation name="LWJDBC - InsertCustomer">
        <participant name="ExampleLWJDBCBusinessProcess"/>
        <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="number(/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_id/text())"/>
        <assign to="param2" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_name/text()"/>
        <assign to="param3" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_address/text()"/>
        <assign to="param4" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_phone/text()"/>
```

The LWJDBC instance used to determine if the customer exists

The LWJDBC instance used to add a customer to the database

```
        <assign to="paramtype1">Integer</assign>
        <assign to="paramtype2">String</assign>
        <assign to="paramtype3">String</assign>
        <assign to="paramtype4">String</assign>
        <assign to="query_type">ACTION</assign>
        <assign to="result_name">RootResultXmlElement</assign>
        <assign to="row_name">ResultsFromInsert</assign>
        <assign to="sql">INSERT INTO CUSTOMER2 VALUES(?, ?, ?, ?)</assign>
        <assign to="." from="*"/>
    </output>          <input message="inmsg">
        <assign to="." from="."/>
    </input>
    </operation>
    </sequence>
    <sequence name="UpdateCustomer">
    <operation name="LWJDBC - UpdateCustomer">
    <participant name="ExampleLWJDBCBusinessProcess"/>
    <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="param1" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_name/text()"/>
        <assign to="param2" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_address/text()"/>
        <assign to="param3" from="/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_phone/text()"/>
        <assign to="param4" from="number(/ProcessData/CustomerInformation/ResultsFromQuery/
Customer[number(//NumberOfRecords)]/customer_id/text())"/>
        <assign to="paramtype1">String</assign>
        <assign to="paramtype2">String</assign>
        <assign to="paramtype3">String</assign>
        <assign to="paramtype4">Integer</assign>
        <assign to="query_type">ACTION</assign>
        <assign to="result_name">RootResultXmlElement</assign>
        <assign to="row_name">ResultsFromUpdate</assign>
        <assign to="sql">UPDATE CUSTOMER2 SET CUSTOMER_NAME = ?, CUSTOMER_ADDRESS=?,
CUSTOMER_PHONE=? where CUSTOMER_ID=?</assign>
        <assign to="." from="*"/>
    </output>
    <input message="inmsg">
        <assign to="." from="."/>
    </input>
    </operation>
    </sequence>
    </choice>
    <assign to="NumberOfRecords" from="NumberOfRecords - 1"/>
    <repeat name="CheckMoreRecords" ref="ContinueProcessing"/>
    </sequence>
    </choice>
  </sequence>
</process>
```

The LWJDBC instance used to update an existing customer's information in the database

For information about the Lightweight JDBC parameters, see *Implementing the Lightweight JDBC Adapter*. For additional Lightweight JDBC adapter examples, including examples for using Oracle stored procedures and functions, see *Lightweight JDBC Business Process Usage*.

## Implementing the Lightweight JDBC Adapter

You can implement the Lightweight JDBC adapter by:

- Configuring a Lightweight JDBC adapter instance to include in business processes. This lets you reuse the same Lightweight JDBC adapter instance to run different SQL queries against a database.
- Configuring a Lightweight JDBC instance that runs an SQL query and then starts a business process.

The information in this section applies to both implementation types.

To implement the Lightweight JDBC adapter, complete the following tasks:

1. Collect the following information:
   - The name of the business process (if the adapter is to start a business process)

- Database pool name used to connect to an external database
- Parameter values for the jdbc_customer.properties.in file (see *Adding New Database Pools*).

2. Set up a connection to an external database, if you have not already done so. For information, see *Setting Up a Connection to an External Database*.

3. Create a Lightweight JDBC adapter configuration. For information, see *Managing Services and Adapters*.

4. Configure the Lightweight JDBC adapter. For information, see *Configuring the Lightweight JDBC Adapter* .

5. Use the Lightweight JDBC adapter in a business process.

   **Note:** If the Lightweight JDBC adapter configuration starts a business process, create the business process prior to configuring the adapter.

## Setting Up a Connection to an External Database

You must set up a connection to an external database for the Lightweight JDBC adapter. You can use any of the databases supported by the application for internal use or other JDBC-compliant databases, such as Sybase. For information about databases supported by the application, see *System Requirements*.

## Adding New Database Pools

To define a new database pool for use by the Lightweight JDBC adapter, you must add settings for the pool to the jdbc_customer.properties.in file, which is located in the application's */install_dir/*properties directory.

In jdbc_customer.properties.in, specify the database server name, port number, database/catalog name, user ID and password. To encrypt your database password, use the encrypt_string.sh or encrypt_string.cmd utility in the bin directory. Then place the encrypted password, prefixed by an encryption indicator, in your properties file.

**CAUTION:**
**There are two jdbc_customer.properties files: jdbc_customer.properties.in, which is the "template" properties file; and jdbc_customer.properties, which is the "packaged" properties file. It is extremely important to ensure that you add the records to the template file, jdbc_customer.properties.in, not to the packaged file. Each time you run the setupfiles command in the application, all packaged files are updated with the information contained in their template (.in) files. This means that if you make changes to the packaged file, jdbc_customer.properties, they are lost each time setupfiles is run. Always make changes to the template file, jdbc_customer.properties.in, and your changes will be maintained.**

If the database you want to connect to resides on a database server type that is not the same as the application's database server type, you also need to install a JDBC driver using the install3rdparty.sh or install3rdparty.cmd utility.

Select a table and column in your database to use in the test on reserve function. This function causes the application to test the database connection using a quickly run query before attempting to use it. This function ensures that idle connections are revived. The column referenced in the query should be of the type varchar and should be at least five characters in length.

## Connecting to an External Database

To connect to an external database:

1. Add the necessary records to the jdbc_customer.properties.in file found in the /install_dir/properties directory.

   **Note:** If invalid data (like ABC or 13.45) is entered in a pool setting, the setting uses its default value.

   See the examples that follow this procedure for *Oracle 8i/9i*, *DB2*, *MS SQL 2000*, and *Sybase*.

   The following table contains the parameters needed to add a new database pool to the jdbc_customer.properties.in file:

| Parameter | Description |
|---|---|
| *databasePool*.driver | JDBC driver class file for the database application |
| *databasePool*.url | Database location (full URL as defined by the Java JDBC standards).<br>**Note:** For Oracle systems, the last segment in the URL is the Oracle SID (not the System Reference or Tnsnames entry).<br>**Note:** You can locate the Java JDBC standards at http://www.java.sun.com. |
| *databasePool*.user | Username for logging into the database |
| *databasePool*.password | Password for logging into the database |
| *databasePool*.maxconn | Maximum number of database connections for the connection pool. |
| *databasePool*.storedProcClassName | Specifies the class that handles stored procedure calls for the Lightweight JDBC adapter. The following classes are used for the database types:<br>• MSSQL, Sybase, and DB2 –<br><br>  com.sterlingcommerce.woodstock. util.frame.jdbc.GenericStoredProcQuery<br>• Oracle 8i/9i –<br><br>  com.sterlingcommerce.woodstock. util.frame.jdbc.OracleNoAppStoredProcQuery<br><br>**Note:** The Lightweight JDBC adapter does not support stored procedures for MySQL. |

| Parameter | Description |
|---|---|
| *databasePool*.varDataClassName | Each database that the application supports handles binary objects differently. This parameter specifies the class used to handle binary data for the database. Enter the correct class for your database:<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.DB2ISeriesVarData<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.DB2VarData<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.DB2ZOSVarData<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.JConnectVarData<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.MSSQLVarData<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.MySQLVarData<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.OracleBlobVarData<br>• com.sterlingcommerce.woodstock. util.frame.jdbc.OracleVarData |
| *databasePool*.catalog | Database name (usually the same as the last segment of the URL) |
| *databasePool*.type | Valid values:<br>• local<br>• remote (default) |
| *databasePool*.testOnReserve | Whether to test the connection. Valid values:<br>• true<br>• false<br>**Note:** This function causes the application to test the database connection before attempting to use it and revives idle connections. |
| *databasePool*.testOnReserveQuery | SQL query to use when testing the connection . Select a table and column in your database to use in the test on reserve function. The column referenced in the query should be of the type varchar and should be at least five characters in length. This query needs to be executable by the *databasePool*.username account and must be a valid SQL query.<br><br>For example: SELECT table_name FROM user_tables WHERE table_name=?<br><br>Where ? must accept a string value. The query does not have to return a value to operate. If the query fails, the Database Pool is not activated. |

| Parameter | Description |
|---|---|
| *databasePool.testOnReserveInterval* | The minimum number of milliseconds between running testOnReserve on the same connection. The default value is 60000. Valid values:<br>• 0 - No interval and current interval is used.<br>• <= 0 - No interval.<br>• > 0 - The minimum number of milliseconds between running testOnReserve on the same connection. |
| *databasePool*.max8177RetryCount | Only used for an Oracle database, this tells the software how many times to retry if it receives an ORA-8177 error in certain situations. |
| *databasePool*.dbvendor | Enter the database name: sybase, oracle, mysql, mssql, db2, db2zos, db2iseries, or other vendor name. |
| *databasePool*.maxsize | Maximum size of the database pool. This property was previously contained in the poolManager.properties file. This value must not exceed the value specified for the databasePool.maxconn parameter in the jdbc.properties file. |
| *databasePool*.initsize | Initial size of the database pool. This property was previously contained in the poolManager.properties file.<br><br>**Note:** The best practice is to set the databasePool.initsize to 0 (zero). The databasePool.initsize controls the number of connections that are created for the pool during startup. By setting the databasePool.initsize to 0 (Zero), it allows the pool to be created, but will not create any connections at the startup time. This allows the system to start even if the external database is not available. When a connection is needed from the pool, one will be created at that time. Connections will still be pooled up to whatever the databasePool.maxconn is set to. |
| *databasePool*.factory | Always enter the following:<br><br>com.sterlingcommerce.woodstock. util.frame.jdbc.ConnectionFactory<br>. |

| Parameter | Description |
|---|---|
| *databasePool*.behaviour | Behavior a connection pool exhibits when it runs out of connections. This property replaces the databasePool.onEmpty property in the former poolManager.properties file. Valid values: |
| | • 0 – The pool simply returns indicating to the software to abort its current action and try again later. This value corresponds to the value return in the databasePool.onEmpty property. |
| | • 1 – The pool waits the number of milliseconds specified in databasePool.waittime for a connection to be returned before indicating to the software to abort and try again. This value corresponds to the value wait in the databasePool.onEmpty property. |
| | • 2 – The pool creates a buffered connection (a connection above the size specified in databasePool.maxsize). When using a setting of 2, the maximum number of connections for the pool is the value specified for databasePool.maxsize plus the value specified for databasePool.buffersize connections. This allows connections to be created under heavy demand. This value corresponds to the value new in the databasePool.onEmpty property. |
| *databasePool.lifespan* | The number of milliseconds a connection will live in a given pool before it needs to be removed. |
| | • 0 – (Default) No timeout |
| | • <= 0 – No timeout |
| | • > 0 – Number of milliseconds that a connection stays in pool |
| *databasePool.idletimeout* | The number of milliseconds a connection can stay idle in a given pool before it needs to be removed. The default value is 86400000. Valid values: |
| | • 0 – No timeout |
| | • <= 0 – No timeout |
| | • > 0 – Number of milliseconds that a connection stays in pool |
| *databasePool.housekeepinginterval* | The minimum number of milliseconds between running the housekeeping task to clean out idle connections. Valid values are any positive number. The default value is 3600000 milliseconds (1 hour). Any number less than 3600000 will cause the default of 3600000 milliseconds to be used. |
| *databasePool*.buffersize | Number of extra connections that the connection pool can create above the value specified for databasePool.maxsize to improve handling of unanticipated loads on the system. This property is only used if databasePool.behavior is set to 2. |
| *databasePool*.waittime | Amount of time (in milliseconds) to wait for a connection to become available before indicating to the software to abort the current action and try again later. This property is only used if databasePool.behavior is set to 1. |

| Parameter | Description |
|---|---|
| databasePool.transaction | Determines whether the dbpool can participate in distributed transactions. Valid values: true – if true, the databasePool.type value should be remote or should not be declared. false (default). |
| | The following pools can not participate in distributed transactions at this time: Local Pool, noTrans Pool, and Yantra Pool. |

2. Run the setupfiles.sh (UNIX) or setupfiles.cmd (Windows) utility located in the /install_dir/bin directory of the application installation directory. This updates the "packaged" properties file, jdbc_customer.properties, with the changes from the "template" properties file, jdbc_customer.properties.in.

3. If the vendor for the connection database is not the same vendor as the database vendor used for the application's database, install the appropriate JDBC driver to access the database server. Use the install3rdParty.sh (UNIX) or install3rdparty.cmd (Windows) utility located in the /install_dir/bin directory of the application's installation directory to add the JDBC driver jar files. Type install3rdParty on the command line to get a description of the parameters you can specify.

   The following examples are for a UNIX environment. The vendor name and version are the first two parameters, along with the location of the zip file containing the JDBC driver files.

   - For Oracle 9i, install the driver using the following command:

     ./install3rdParty.sh Oracle 9_2_0_5 -d *usr/local/directory*/oracle/9_2_0_5/classes12.zip

   - For DB2, install the driver using the following command:

     ./install3rdParty.sh db2java 7_2 -d *usr/local/directory*/db2java.zip

   - To install the jConnect driver for Sybase, refer to *Installing a Sybase Driver*.

4. Stop and restart the application to use the changed files.

## Installing a Sybase Driver

Install the jConnect driver for Sybase using the following procedure:

1. Download jConnect-5_5.zip from the Sybase web site.
2. Run the following command:

   ./install3rdParty.sh jconnect 5_5 -d *usr/local/directory*/5_5/jConnect-5_5.zip

   - If this command succeeds, you are finished with this procedure.
   - If the application reports in the system log that the driver could not be registered because the driver class cannot be found, continue with the procedure. Use the following steps to remove existing references to jConnect.

3. Stop the application.
4. Change your directory to *install_dir*/jar.
5. Delete any existing folders referencing jConnect.
6. Change your directory to *install_dir*/properties.
7. Open the files dynamicclasspath.cfg and dynamicclasspath.cfg.in. Delete any lines referencing jConnect, and save the files.
8. Create the following temporary directory:

*install_dir*/bin/jconnect

9. Extract only the jar files from jConnect-5_5.zip to this directory.

10. Run the following command:

```
install3rdparty jconnect 5_5 -d install_dir/bin/jconnect/*.jar
```

11. Check *install_dir*/jar/jconnect/5_5/*your_platform* to make sure that six jar files have been copied successfully.

12. Open the dynamicclasspath.cfg file in *install_dir*/properties and check that the following entries are there:

```
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jconn2.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jconn2d.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jisql.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/jTDS2d.jar
VENDOR_JAR=/install_dir/jar/jconnect/5_5/your_platform/ribo.jar
```

13. Open the customer.jdbc.properties.in file in *install_dir*/properties and check that the following entries are there:

```
jconnectPool.driver=com.sybase.jdbc2.jdbc.SybDriver
jconnectPool.url=jdbc:sybase:Tds:your Hostname:4100/your DB
jconnectPool.user=your user name
jconnectPool.password=your password
jconnectPool.catalog=your database
jconnectPool.type=local
jconnectPool.testOnReserve=true
jconnectPool.testOnReserveQuery=your Test On Reserve Query
jconnectPool.dbvendor=Sybase
jconnectPool.buffersize=500
jconnectPool.maxsize=28
jconnectPool.initsize=1
jconnectPool.behaviour=2
jconnectPool.waittime=1000
jconnectPool.storedProcClassName= com.sterlingcommerce.woodstock.util.
     frame.jdbc.SybaseStoredProcQuery
jconnectPool.varDataClassName=com.sterlingcommerce.woodstock.util.
     frame.jdbc.JConnectVarData
jconnectPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
     ConnectionFactory
```

14. Save the customer.jdbc.properties.in file and run the following command:

```
install_dir/bin/setupfiles.sh
```

This procedure should result in a successful connection to your Sybase database. However, if the database has been configured as character set ROMAN8, it is likely that you will see the following message in the application system log, because of a limitation in the Sybase driver:

```
java.sql.SQLWarning: JZ0IB: The server's default
charset of roman8 does not map to an encoding that is available
in the client Java environment. Because jConnect will not be able
to do client-side conversion, the connection is unusable and is
being closed. Try using a later Java version or try including your
Java installation's i18n.jar or charsets.jar file in the clasps
```

One resolution of this issue is to configure the primary Adaptive Server with a default character set that maps to one of the character sets supported by jConnect for JDBC (for example, UTF-8). For more details, refer to the Sybase documentation.

Another resolution of this issue is to use the open source jTDS driver from Sourceforge (sourceforge.net). To install this driver, follow these instructions:

1. Stop the application.

2. Remove references to jConnect as described previously.

3. Copy the jtds-1.2.jar file to an accessible directory on the application machine.

4. Run the following command:

   *install_dir*/bin/Install3rdparty.sh jTDS 1_2 - jar *absolutePath*/jtds-1.2.jar

5. Check that the dynamicclasspath.cfg file has picked up this change. For example, */install_dir*/jar/jTDS/1_2/*your_platform*/jtds-1.2.jar.

6. Edit the jdbc_customer.properties.in file. The definition of the pool should be similar to the following example:

```
jTDSPool.driver=net.sourceforge.jtds.jdbc.Driver
jTDSPool.url=jdbc:jtds:sybase://your
Hostname:4100/answer
jTDSPool.user=your user name
jTDSPool.password=your password
jTDSPool.catalog=your database
jTDSPool.type=local
jTDSPool.testOnReserve=false
#jTDSPool.testOnReserveQuery=your Test On Reserve Query
jTDSPool.dbvendor=jtds
jTDSPool.buffersize=50
jTDSPool.maxsize=20
jTDSPool.initsize=5
jTDSPool.behaviour=2
jTDSPool.storedProcClassName=com.sterlingcommerce.woodstock.util.
      frame.jdbc.SybaseStoredProcQuery
jTDSPool.varDataClassName=com.sterlingcommerce.woodstock.util.
      frame.jdbc.GenericVarData
jTDSPool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
      ConnectionFactory
```

7. Restart the application.

## Properties File Examples for Specific Databases

### Oracle 8i/9i

For Oracle 8i/9i, enter the following parameters in the jdbc_customer.properties.in file, where *databasePool* is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=oracle.jdbc.driver.OracleDriver
databasePool.url=jdbc:oracle:thin:@servername:0000:servername
databasePool.user=username
databasePool.password=password
databasePool.catalog=catalogname
databasePool.type=local
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from
    Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.max8177RetryCount=n
databasePool.dbvendor=oracle
databasePool.buffersize=n
databasePool.maxsize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=n databasePool.storedProcClassName=
    com.sterlingcommerce.woodstock.util.frame.jdbc.OracleNoAppStoredProcQuery
```

```
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.
    frame.jdbc.OracleVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
    ConnectionFactory
```

## DB2

For DB2, enter the following parameters in the jdbc_customer.properties.in file, where databasePool is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

**Note:** The JDBC adapter does not support stored procedures for DB2/iSeries and DB2/zOS.

```
databasePool.driver=com.ibm.as400.access.AS400JDBCDriver
databasePool.url=jdbc:as400://serverName/DB2Database;translate binary=true;
    transaction isolation=none;
databasePool.catalog=DB2Database
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.frame.
    jdbc.DB2ISeriesVarData
databasePool.user=username
databasePool.password=password
databasePool.maxconn=20
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT SI_VERSION from SI_VERSION where
    SI_VERSION = ?
databasePool.testOnReserveInterval=60000
databasePool.blobPageSize=1024000
databasePool.compressBlob=true
databasePool.cacheps=true
databasePool.dbvendor=db2iSeries
databasePool.buffersize=500
databasePool.maxsize=20
databasePool.initsize=0
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
    ConnectionFactory
databasePool.behaviour=2
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=1000
```

## MS SQL 2000

For MS SQL 2000, enter the following parameters in the jdbc_customer.properties.in file, where databasePool is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=com.microsoft.jdbc.sqlserver.SQLServerDriver
databasePool.url=jdbc:microsoft:sqlserver://servername:0000;
    DatabaseName=SQLdatabase;SelectMethod=cursor
databasePool.user=username
databasePool.password=password
databasePool.catalog=catalogname
databasePool.type=local
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from
    Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.dbvendor=mssql
databasePool.buffersize=n
```

```
databasePool.maxsize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=n
databasePool.storedProcClassName=com.sterlingcommerce.woodstock.util.
     frame.jdbc.GenericStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.
      frame.jdbc.MSSQLVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
     ConnectionFactory
```

## Sybase

For Sybase, enter the following parameters in the jdbc_customer.properties.in file, where databasePool is the name of the pool you are adding. Sample values are italicized; enter the correct value for your environment instead of the sample value. Values that are not italicized are the actual values that you should enter for the parameter:

```
databasePool.driver=com.sybase.jdbc2.jdbc.SybDriver
databasePool.url=jdbc:sybase:Tds:servername:0000/SybaseDB
databasePool.user=username
databasePool.password=password
databasePool.catalog=catalogname
databasePool.type=local
databasePool.testOnReserve=true
databasePool.testOnReserveQuery=SELECT TestConnection from
     Connection_tb WHERE TestConnection = ?
databasePool.testOnReserveInterval=60000
databasePool.dbvendor=Sybase
databasePool.buffersize=n
databasePool.maxsize=n
databasePool.initsize=n
databasePool.behaviour=n
databasePool.lifespan=0
databasePool.idletimeout=86400000
databasePool.housekeepinginterval=3600000
databasePool.waittime=n
databasePool.storedProcClassName=com.sterlingcommerce.woodstock.util.
     frame.jdbc.SybaseStoredProcQuery
databasePool.varDataClassName=com.sterlingcommerce.woodstock.util.
     frame.jdbc.JConnectVarData
databasePool.factory=com.sterlingcommerce.woodstock.util.frame.jdbc.
     ConnectionFactory
```

## Encrypting Your Database Password

To use encryption for the database password:
1. Use encrypt_string.sh (UNIX) or encrypt_string.cmd (Windows).
2. When prompted, enter your external database password.

   The script returns the encrypted value for your password.
3. Place the encrypted password in your jdbc_customer.properties.in file entry (see step 2 in *Connecting to an External Database*), prefixing the encrypted password with ENCRYPTED.

   For example, *databasePool*.password=ENCRYPTED:rO0ABXQABkRFU2VkZXVy

## Configuring the Lightweight JDBC Adapter

To configure the Lightweight JDBC adapter, you must specify field settings in the application and in the GPM.

### Application Configuration Parameters

The following table describes the fields used to configure the Lightweight JDBC adapter:

| Field | Description |
|---|---|
| Name (Config) | Unique and meaningful name for the adapter configuration. Required. |
| Description | Meaningful description for the adapter configuration, for reference purposes. Required. |
| Select a Group | Select one of the options:<br>• None – You do not want to include this configuration in a group at this time.<br>• Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration.<br>• Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. |
| Start a new business process (StartNewWorkFlow) | Whether to start a new business process . Valid values:<br>• This Lightweight JDBC adapter will start a new business process.<br>• This Lightweight JDBC adapter will not start a new business process. |
| Run as User | Enter (or select from the list) the user ID to be associated with business process instances of this service. |
| Use 24 Hour Clock Display | Select to specify times for this schedule using the 24 hour clock. Leave blank to use 12 hour clock and AM and PM. |

| Field | Description |
|---|---|
| Schedule | Information about scheduling the Lightweight JDBC adapter configuration to run and to start the specified business process. Valid values: |
| | • Do not use schedule |
| | If this field is selected, this service does not start a business process and does not run on a schedule. |
| | • Run based on timer |
| | Valid values are the hour and minutes at which to run the service. Indicate whether you want the service to run at startup. |
| | • Run daily |
| | Valid values are the hour and minutes at which to run the service, daily. You can also specify a time interval. Indicate whether you want the service to run at startup. |
| | • Run based on day(s) of the week |
| | Valid values are the day of the week, the hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. |
| | • Run based on day(s) of the month |
| | Valid values are the day of the month (including the last day of the month (LDOM)), hour, and the minutes at which to run the service. You can also specify a time interval. Indicate whether you want the service to run at startup. |
| | **Note:** The Schedule field only displays as an option if you set the *Start a new business process* parameter to "This Lightweight JDBC adapter configuration starts a new business process." |
| Business process (InitialWorkFlowId) | Business process you want the Lightweight JDBC adapter to start, if any. Required if you set the *Start a new business process* parameter to "This Lightweight JDBC adapter configuration will start a new business process." |
| Pool Name | Select the database pool to be used. |
| XML Result Root Tag (result_name) | The root tag element you want to appear in the XML document returned from the Lightweight JDBC adapter after a query. This document contains the results from the SQL query and becomes the primary document. For example, if you specify RootResultXmlElement as the name for the XMLResult Root Tag, the following results are displayed in the returned XML document for an ACTION query: |
| | **\<RootResultXmlElement\>**\<ResultsFromUpdate\><br>\<Rows_Affected\><br>1\</Rows_Affected\><br>\</ResultsFromUpdate\><br>**\</RootResultXmlElement\>** |

| Field | Description |
|---|---|
| XML Result Row Tag (row_name) | Row tag XML element containing all the column tags returned from the SQL query. The column tag names are not configurable. The tag names are generated by the column name returned in the result set. Required.<br><br>For example, if you specify `ResultsFromUpdate` as the name for the XML Result Row Tag, the following results are displayed in the returned XML document for an ACTION query.<br><br>`<RootResultXmlElement>`<br>**`<ResultsFromUpdate>`**`<Rows_Affected>`<br>`1</Rows_Affected>`<br>**`</ResultsFromUpdate>`**`</RootResultXmlElement>` |
| Query Type (query_type) | Result set or the number of rows affected by an action query returned by the SQL statement. Required. Valid values:<br><br>• SELECT – Returns the results from a query. The results are bounded to the XML Result Root tag (result_name), and then each returned row is bounded to the XML Result Row Tag (row_name). Each column from the Select statement also forms an XML element with the element name matching the column name and the element data matching the data returned from the query.<br><br>For example, if you query for all customers in a database (SELECT * FROM Customer), the results returned might look like the following:<br><br>`<XMLResultRootTag>`<br>`<XMLResultRowTag>`<br>`<CUSTOMER_ID>1</CUSTOMER_ID>`<br>`<CUSTOMER_NAME>Jane Doe</CUSTOMER_NAME>`<br>`</XMLResultRowTag>`<br>`<XMLResultRowTag>`<br>`<CUSTOMER_ID>2</CUSTOMER_ID>`<br>`<CUSTOMER_NAME>Joe User</CUSTOMER_NAME>`<br>`</XMLResultRowTag>`<br>`</XMLResultRootTag>`<br><br>• INSERT – Executes an insert query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above.<br><br>• UPDATE – Executes an update query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above.<br><br>• ACTION – Executes an INSERT, UPDATE, or DELETE query and returns the number of rows affected. See the example for the XML Result Row tag parameter provided with SELECT, above.<br><br>• Stored procedure/function – Applies to stored procedures and functions. |

| Field | Description |
|---|---|
| | Notes for stored procedures and functions: |
| | • When defining a Lightweight JDBC adapter instance using the GPM, you must declare a Query type of Stored Procedure/Function. When defining a Lightweight JDBC adapter instance using the BPML, you must declare a Query type of PROCEDURE. |
| | • You should specify the stored procedure on the SQL parameter in the Lightweight JDBC adapter instance definition using the JDBC syntax. For example: |
| | `{call StoredProcedureName(?,?,?)}` |
| | where ?,?,? are the arguments passed in from the parameter specifications (param1-param20) defined in the Lightweight JDBC adapter instance definition. |
| | • You should specify the function on the SQL parameter in the Lightweight JDBC adapter instance definition using the JDBC syntax. For example: |
| | `{? = call product_count()}` |
| | where ? is the argument returned from the function. You must specify a parameter type that matches the format of the value returned from the function. For example: |
| | `<assign to="paramtype1">Integer</assign>` |
| | • There are limitations when using Oracle to call stored procedures. For information about these limitations, see *Calling Oracle Stored Procedures*. |
| SQL Statement (sql) | Hard-coded SQL query that queries a database. It must be in valid SQL syntax.<br>**Note:** You can specify the SQL query either using this field or when you add the Lightweight JDBC adapter instance in a business process. |

## GPM Configuration

The following screen shows the graphical view of the GPM parameters for the Lightweight JDBC adapter. Values for the inactive fields were specified using the Lightweight JDBC adapter configuration. The active fields cannot be configured in the application or are being overridden. None of the fields on the Message From Service tab require configuration.

| Name | | |
|---|---|---|
| Name | LWJDBC - QueryForCustomerID | |
| Config. | ExampleLWJDBCBusinessProcess | |

**Message To Service** | Message From Service

| Output Msg | Obtain Message first, then Process Data |
|---|---|
| Message Name | LightweightJDBCAdapterTypeInputMessage |

| Name | Value | Use XPAT |
|---|---|---|
| InitialWorkFlowId | | ☐ |
| param1 | number(/ProcessData/CustomerInformation/ResultsFromQuery/Customer[number(//NumberOfRecords)]/customer_id/text()) | ☑ |
| param10 | | ☐ |
| param11 | | ☐ |
| param12 | | ☐ |
| param13 | | ☐ |
| param14 | | ☐ |
| param15 | | ☐ |
| param16 | | ☐ |
| param17 | | ☐ |
| param18 | | ☐ |
| param19 | | ☐ |
| param2 | | ☐ |
| param20 | | ☐ |
| param3 | | ☐ |
| param4 | | ☐ |
| param5 | | ☐ |
| param6 | | ☐ |
| param7 | | ☐ |
| param8 | | ☐ |

| Name | Value | Use XPATH? |
|---|---|---|
| param9 | | ☐ |
| paramtype1 | Integer | ☐ |
| paramtype10 | | ☐ |
| paramtype11 | | ☐ |
| paramtype12 | | ☐ |
| paramtype13 | | ☐ |
| paramtype14 | | ☐ |
| paramtype15 | | ☐ |
| paramtype16 | | ☐ |
| paramtype17 | | ☐ |
| paramtype18 | | ☐ |
| paramtype19 | | ☐ |
| paramtype2 | | ☐ |
| paramtype20 | | ☐ |
| paramtype3 | | ☐ |
| paramtype4 | | ☐ |
| paramtype5 | | ☐ |
| paramtype6 | | ☐ |
| paramtype7 | | ☐ |
| paramtype8 | | ☐ |

| | | |
|---|---|---|
| paramtype9 | | ☐ |
| pool | mysqlTrainingPool | ☐ |
| query_type | Select | ☐ |
| result_name | RootResultXmlElement | ☐ |
| row_name | ResultsFromSQL | ☐ |
| sql | SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2... | ☐ |
| StartNewWorkFlow | This Lightweight JDBC Adapter will not start a new busine... | ☐ |

The following example shows the corresponding BPML parameters for the
Lightweight JDBC adapter GPM parameters. This example business process takes
the primary document, which contains the query results, and writes it to process
data.

```
<process name="ExampleLWJDBC">
<operation name="LWJDBC - QueryForCustomerID">
<participant name="ExampleLWJDBCBusinessProcess"/>
<output message="LightweightJDBCAdapterTypeInputMessage">
<assign to="param1" from="number(/ProcessData/CustomerInformation/
     ResultsFromQuery/Customer[number(//NumberOfRecords)]/customer_id/text())"/>
<assign to="paramtype1">Integer</assign>
<assign to="query_type">SELECT</assign>
<assign to="sql">SELECT COUNT(*) AS CUSTOMERS FROM CUSTOMER2 WHERE
     CUSTOMER_ID = ?</assign>
<assign to="." from="*"/>
```

```
</output>
<input message="inmsg">
<assign to="NumberOfCustomers" from="DocToDOM(PrimaryDocument)/
     ResultsFromSQL/CUSTOMERS"/>
</input>
</operation>
</sequence>
</process>
```

The following table describes the fields used to configure the Lightweight JDBC adapter in the GPM. This table contains only the fields that are configured in the GPM. See *Configuring the Lightweight JDBC Adapter* for parameters you can specify in either the GPM or in the configuration for your application.

| Field | Description |
| --- | --- |
| Config | Name of the service configuration. |
| param1 - param20 | Specifies the values that are passed to the SQL statement that is to be executed against the database. You can either specify constant values or use XPath query statements to retrieve information from process data to pass to the SQL statement. If you use XPath query statements, you must also select the "Use XPath?" check box in the Service Editor of the GPM for the parameter. See the *Lightweight JDBC Business Process Usage* section for an example of using Xpath to set these values. Optional. |
| paramtype1 - paramtype20 | Parameter type that corresponds to the parameter number (for example, param1 and paramtype1). Every parameter specified must have a corresponding parameter type. Valid values: <br> • Cursor <br> • String <br> • CharacterStream <br> • CharacterStreamFromDocument <br> • Integer <br> • Long <br> • Double <br> • Float <br> • Date |

## Large Text Data Support

Two values are available for large text data (clob) support:
* **CharacterStream** is used to insert or update data to the database that is in process data.
* **CharacterStreamFromDocument** is used to insert or update data to the database from the primary document.

To insert data to the database or to update the database using the CharacterStream or CharacterStreamFromDocument values, there are two new query types: INSERT and UPDATE.

There is also a parameter that you can add to your business process manually (this parameter is not available through the GPM): write_characterstream_to_document. Valid values for this parameter are YES and NO. This parameter enables you to

place the contents of large data inside of a document during a SELECT query, because XML has size limitations on pcdata. The primary document will still contain the result tree but the name of the document will replace the large data. This new parameter is optional and defaults to NO for backward compatibility.

**Note:** When inserting, updating, or selecting clobs from Oracle, ensure the varDataClassName in the jdbc.properties.in file is pointing to OracleBlobVarData for an external pool. The Lightweight JDBC adapter will attempt to get a connection a set number of times. The number of retries is configurable by changing the externalDBPoolRetries property in the noapp.properties.in file.

The following example illustrates using the parameter to write the contents of the data returned from the database directly to a document:

```
<operation name="LightweightJDBCAdapterType">
     <participant  name="LightweightJDBCAdapterQuery"/>
     <output message="LightweightJDBCAdapterTypeInputMessage">
       <assign to="result_name">result</assign>
       <assign to="sql">SELECT ID, XML_MESSAGE FROM CUSTOMER WHERE
           CUSTOMER_ID=?</assign>
       <assign to="pool">mysqlPool_local</assign>
       <assign to="param1" from="customer_id/text()" />
       <assign to="paramtype1">String</assign>
       <assign to="write_characterstream_to_document">Yes</assign>
       <assign to="row_name">row</assign>
       <assign to="query_type">SELECT</assign>
       <assign to="." from="*"></assign>
     </output>
     <input message="inmsg">
       <assign to="." from="*"></assign>
     </input>
   </operation>
```

This is a sample of what you would see in the primary document:

```
<?xml version="1.0" encoding="UTF-8"?>
    <result>
     <row>
      <ID>1</ID>
      <XML_MESSAGE>doc-1</XML_MESSAGE>
     </row>
    </result>
```

The parameter types are used by the Lightweight JDBC adapter to insert or update large text data into the database. The insert and update statements differ in how they are implemented, so you must specify the INSERT or UPDATE action type. Several examples follow.

### Example 1

Updates the column with the contents of the primary document. In this example, the parameter type is CharacterStreamFromDocument, so the parameter value is not required; instead, the value is taken from the primary document.

```
<operation name="LightweightJDBCAdapterType">
   <participant name="LightweightJDBCAdapterQuery"/>
   <participant name="LightweightJDBCAdapterQuery"/>
     <output message="LightweightJDBCAdapterTypeInputMessage">
     <output  message="LightweightJDBCAdapterTypeInputMessage">
       <assign to="result_name">resul</assign>t<assign to="result_name">
           result
       </assign>
       <assign to="sql">UPDATE CUSTOMER SET XML_MESSAGE=? WHERE
           CUSTOMER_ID=</assign>?<assign to="sql">
```

```
                UPDATE CUSTOMER SET XML_MESSAGE=?
                WHERE CUSTOMER_ID=?</assign>
      <assign to="paramtype1">CharacterStreamFromDocumen</assign>t<assign to
              ="paramtype1">CharacterStreamFromDocument</assign>
      <assign to="param2" from="customer_id/text()" /><assign to="param2"
              from="customer_id/text()" />
      <assign to="paramtype2">Strin</assign>g<assign to="paramtype2">
            String
      </assign>
      <assign to="pool">mysqlPool_loca</assign>l<assign to="pool">
            mysqlPool_local
      </assign>
      <assign to="row_name">ro</assign>w<assign to="row_name">row</assign>
      <assign to="query_type">UPDAT</assign>E<assign to="query_type">
            UPDATE
      </assign>
      <assign to="." from="*"</assign>><assign to="." from="*"></assign>
   </output></output>
  <input message="inmsg"><input message="inmsg">
      <assign to="." from="*"</assign>><assign to="." from="*"></assign>
   </input></input>
</operation>
```

## Example 2

Updates a column with the string returned from an Xpath query against process
data. In this example, the CharacterStream parameter type is used, so the
parameter value is required.

```
<operation name="LightweightJDBCAdapterType">
      <participant name="LightweightJDBCAdapterQuery"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="result_name">result</assign>
        <assign to="sql">UPDATE CUSTOMER SET XML_MESSAGE = ?
              WHERE CUSTOMER_ID=?</assign>
        <assign to="paramtype1">CharacterStream</assign>
        <assign to="param1" from="rootNode/SampleText2/text()"></assign>
        <assign to="param2" from="customer_id/text()" />
        <assign to="paramtype2">String</assign>
        <assign to="pool">mysqlPool_local</assign>
        <assign to="row_name">row</assign>
        <assign to="query_type">UPDATE</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
</operation>
```

## Example 3

Inserts a row with the string returned from an Xpath query against process data.

```
<operation name="LightweightJDBCAdapterType">
      <participant name="LightweightJDBCAdapterQuery"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="result_name">result</assign>
        <assign to="sql">INSERT INTO CUSTOMER (CUSTOMER_ID, XML_MESSAGE)
              VALUES (?, ?) </assign>
        <assign to="param1" from="customer_id/text()" />
        <assign to="paramtype1">String</assign>
        <assign to="paramtype2">CharacterStream</assign>
        <assign to="param2" from="rootNode/SampleText/text()"></assign>
        <assign to="pool">mysqlPool_local</assign>
        <assign to="row_name">row</assign>
        <assign to="query_type">INSERT</assign>
```

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

## Example 4

Writes the contents of the data returned from a database directly to a document (note that this is not the primary document). The primary document will still contain the XML result tree; however, the data in the document is replaced by a document name.

```
<operation name="LightweightJDBCAdapterType">
      <participant name="LightweightJDBCAdapterQuery"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="result_name">result</assign>
        <assign to="sql">SELECT XML_MESSAGE FROM CUSTOMER WHERE
            CUSTOMER_ID=?</assign>
        <assign to="pool">mysqlPool_local</assign>
        <assign to="param1" from="customer_id/text()" />
        <assign to="paramtype1">String</assign>
        <assign to="write_characterstream_to_document">Yes</assign>
        <assign to="row_name">row</assign>
        <assign to="query_type">SELECT</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
</operation>
```

This is an example of what you should expect to see in the primary document:

```
<?xml version="1.0" encoding="UTF-8"?>
    <result>
     <row>
      <XML_MESSAGE>doc-1</XML_MESSAGE>
     </row>
    </result>
```

This is an Oracle example for updating a clob column. The first parameter is used to find the row in the table. The second parameter will be used to update the value.

```
<operation name="LightweightJDBCAdapterType">
      <participant name="LightweightJDBCAdapterQuery"/>
      <output message="LightweightJDBCAdapterTypeInputMessage">
        <assign to="result_name">result</assign>
        <assign to="sql">SELECT XML_MESSAGE FROM CUSTOMER WHERE
            CUSTOMER_ID=?for update</assign>
        <assign to="param1" from="customer_id/text()" />
        <assign to="paramtype1">String</assign>
        <assign to="paramtype2">CharacterStream</assign>
        <assign to="param2" from="rootNode/SampleText2/text()"></assign>
        <assign to="pool">oracleTestPool</assign>
        <assign to="row_name">row</assign>
        <assign to="query_type">UPDATE</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
```

## Lightweight JDBC Business Process Usage

This section contains additional examples using the Lightweight JDBC adapter, including sample configurations for using Oracle stored procedures and functions.

The following sample document is the primary document passed to the business process for the SELECT and ACTION query type examples.

```
<Customer Id="2">
 <Name/>
 <Address/>
 <Phone/>
</Customer>
```

## SELECT Query Type Example

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to SELECT customer information based on the customer ID passed into the business process.
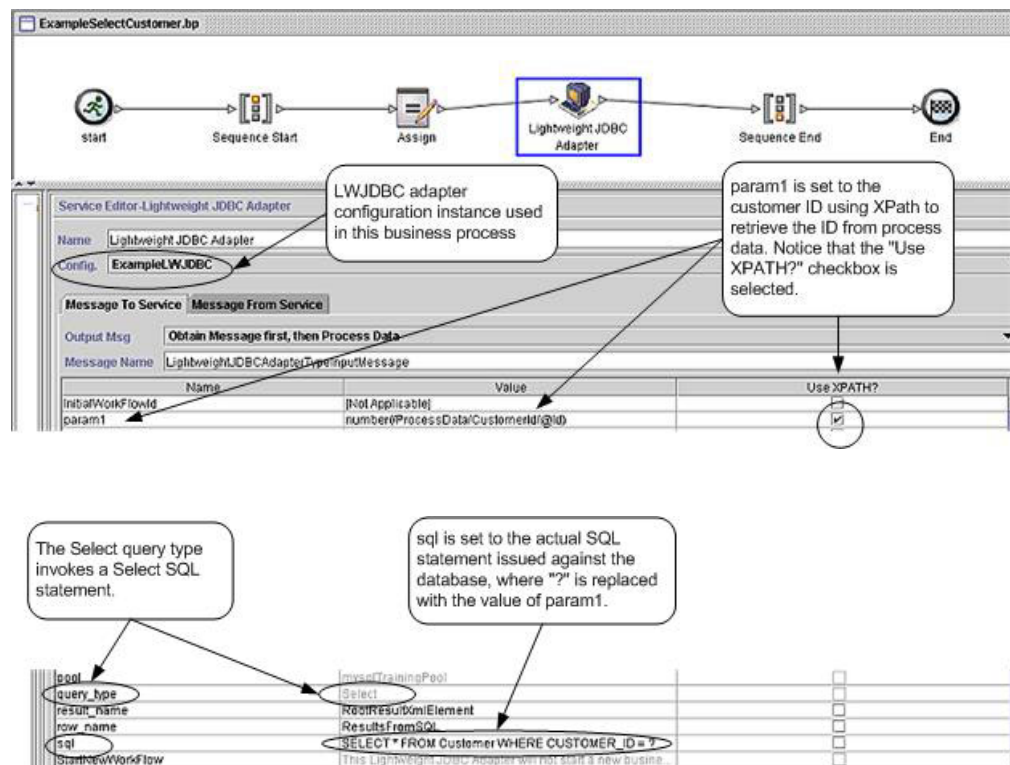


The following example illustrates the same business process using BPML.
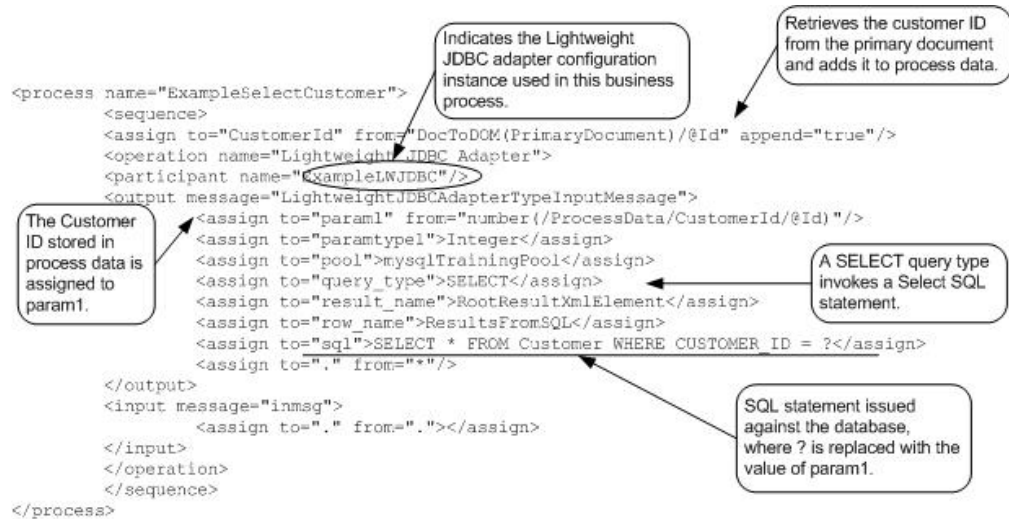
## Message to the Lightweight JDBC Adapter Instance

The following example shows the message sent to the Lightweight JDBC adapter when you run the business process. The message uses the parameters defined within the <output> tags of the Lightweight JDBC adapter definition in the previous BPML example, along with the parameters defined during configuration of the adapter instance.

```
<LightweightJDBCAdapterTypeInputMessage>
 <row_name>ResultsFromSQL</row_name>
 <StartNewWorkFlow>NO</StartNewWorkFlow>
 <result_name>RootResultXmlElement</result_name>
 <pool>mysqlTrainingPool</pool>
 <query_type>SELECT</query_type>
 <XMLVersion>3.0</XMLVersion>
 <param1>2</param1>
 <paramtype1>Integer</paramtype1>
 <sql>SELECT * FROM Customer WHERE CUSTOMER_ID = ?</sql>
</LightweightJDBCAdapterTypeInputMessage>
```

**Note:** You can view the message sent to the Lightweight JDBC adapter from the Business Process Monitor by selecting "Instance Data" at the step where the adapter instance is run and then selecting "Message To Service."

## Message from the Lightweight JDBC Adapter Instance

The following example shows the table definition used to query against the Lightweight JDBC adapter. In the previous BPML example, the assign statement within the <input> tags of the Lightweight JDBC adapter definition (<assign to="." from="."></assign>) indicates that the results are returned as a primary document.

```
create table customer(customer_id NUMERIC not null,
    customer_name varchar(255),
    customer_address varchar(100),
    customer_phone varchar(20));
```

Notice how the columns from the Customer table become XML elements in the results returned from the Lightweight JDBC adapter, as shown below in the message returned from the Lightweight JDBC adapter.

```
<RootResultXmlElement>
        <ResultsFromSQL>
                <customer_id>2</customer_id>
                <customer_name>John Doe</customer_name>
                <customer_address>999 East Lane Denver, CO 80222</customer_address>
                <customer_phone>303-987-5555</customer_phone>
        </ResultsFromSQL>
        <ResultsFromSQL>
                <customer_id>2</customer_id>
                <customer_name>Jane Doe</customer_name>
                <customer_address>999 East Lane Denver, CO 80222</customer_address>
                <customer_phone>303-987-5555</customer_phone>
        </ResultsFromSQL>
</RootResultXmlElement>
```

Corresponds to the value specified for the result_name parameter in the LWJDBC adapter definition.

Corresponds to the value specified for the row_name parameter in the LWJDBC adapter definition.

**Note:** You can view the message returned from the Lightweight JDBC adapter from the Business Process Monitor by selecting "Instance Data" at the step where the adapter instance is run and then selecting "Message From Service."

## ACTION Query Type Example

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to delete customer information from a database.





The following example illustrates the same business process using BPML.

```
<process name="ExampleDeleteCustomer">
        <sequence>
        <assign to="CustomerId" from="DocToDOM(PrimaryDocument)/@Id" append="true"/>
                <operation name="Lightweight JDBC Adapter">
                <participant name="ExampleLWJDBC"/>
                <output message="LightweightJDBCAdapterTypeInputMessage">
                <assign to="param1" from="number(/ProcessData/CustomerId/@Id)"/>
                <assign to="paramtype1">Integer</assign>
                <assign to="query_type">ACTION</assign>
                <assign to="result_name">RootResultXmlElement</assign>
                <assign to="row_name">ResultsFromSQL</assign>
                <assign to="sql">DELETE FROM CUSTOMER WHERE CUSTOMER_ID=?</assign>
                <assign to="." from="*"/>
                </output>
                <input message="inmsg">
                <assign to="." from="."/>
                </input>
                </operation>
        </sequence>
</process>
```

The ACTION query type invokes a Delete SQL statement in this example.

Actual SQL statement issued against the database, where "?" is replaced with the value of param1.

## Message to the Lightweight JDBC Adapter Instance

The following example shows the message sent to the Lightweight JDBC adapter when you run the business process. The message uses the parameters defined within the <output> tags of the Lightweight JDBC adapter definition in the previous BPML, along with the parameters defined for the Lightweight JDBC adapter instance during configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<LightweightJDBCAdapterTypeInputMessage>
 <row_name>ResultsFromSQL</row_name>
 <StartNewWorkFlow>NO</StartNewWorkFlow>
 <result_name>RootResultXmlElement</result_name>
 <pool>mysqlTrainingPool</pool>
 <query_type>ACTION</query_type>
 <XMLVersion>3.0</XMLVersion>
 <param1>2</param1>
 <paramtype1>Integer</paramtype1>
 <sql>DELETE FROM CUSTOMER WHERE CUSTOMER_ID=?</sql>
</LightweightJDBCAdapterTypeInputMessage>
```

**Note:** You can view the message sent to the Lightweight JDBC adapter from the Business Process Monitor by selecting "Instance Data" at the step where the adapter instance is run and then selecting "Message To Service."

## Message from the Lightweight JDBC Adapter Instance

The following example shows the message returned from the Lightweight JDBC adapter to the business process. The assign statement within the <input> tags of the Lightweight JDBC adapter definition (<assign to="." from="."></assign>) indicates that the results are returned as a primary document.

```
<RootResultXmlElement>
<ResultsFromSQL>
<Rows Affected>1</Rows Affected>
</ResultsFromSQL>
</RootResultXmlElement>
```

The Lightweight JDBC adapter returns the number of rows deleted.

**Note:** You can view the message returned from the Lightweight JDBC adapter from the Business Process Monitor by selecting "Instance Data" at the step where the adapter instance is run and then selecting "Message From Service."

## Calling Oracle Stored Procedures

This section describes limitations when using the Lightweight JDBC adapter to call Oracle stored procedures. These limitations apply only to Oracle databases.

- The Lightweight JDBC adapter processes only one result set if you script multiple queries to run in stored procedures.
- The first argument in an Oracle stored procedure must be declared as an output parameter, and the stored procedure must return a value to this parameter even if the value is not needed in the business process.
- You must specify a value for the first parameter (param1) in the Lightweight JDBC adapter instance to correspond with the first argument of the stored procedure, even if the parameter is not used in the stored procedure.

## Stored Procedure Example

The following example shows a database script for creating an Oracle stored procedure. The stored procedure updates the account status for a customer. Notice that the first argument of the stored procedure is declared as an output parameter and is set to a number, even though this value will not be used in the business process.

```
CREATE OR REPLACE PROCEDURE P_Update_Account(v_dummy OUT number,
        v_customer_id IN number, v_account_id number, v_status IN varchar2)
AS
BEGIN
 UPDATE
  ACCOUNT
  SET
   STATUS = v_status
  WHERE
   ACCOUNT_ID = v_account_id
  AND
   CUSTOMER_ID=v_customer_id;
  COMMIT;
  v_dummy:= '4';
END;
/
```

The following sample document is the primary document passed into the example business process used to call the stored procedure.

```
<PurchaseOrder>
 <Account Id="100">
  <Customer Id="2"/>
  <Status>ACTIVE</Status>
 </Account>
</PurchaseOrder>
```

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to start the stored procedure.

ExampleUpdateAccountStatus.bp

start → Sequence Start → Assign → Assign → Assign → Lightweight JDBC Adapter → Sequence End → End

Service Editor-Lightweight JDBC Adapter

Name: Lightweight JDBC Adapter
Config.: ExampleLWJDBCBusinessProcess

Message To Service | Message From Service

Output Msg: Obtain Message first, then Process Data
Message Name: LightweightJDBCAdapterTypeInputMessage

| Name | Value | Use XPATH? |
|---|---|---|
| InitialWorkFlowId | | ☐ |
| param1 | 1 | ☐ |
| param10 | | ☐ |
| param11 | | ☐ |
| param12 | | ☐ |
| param13 | | ☐ |
| param14 | | ☐ |
| param15 | | ☐ |
| param16 | | ☐ |

The value set for param1 is not used in the stored procedure.

| Name | Value | Use XPATH? |
|---|---|---|
| param2 | number(/ProcessData/CustomerId/@Id) | ☑ |
| param20 | | ☐ |
| param3 | number(/ProcessData/Account/@Id) | ☑ |
| param4 | /ProcessData/AccountStatus/text() | ☑ |
| param5 | | ☐ |
| param6 | | ☐ |
| param7 | | ☐ |
| param8 | | ☐ |
| param9 | | ☐ |
| paramtype1 | Integer | ☐ |
| paramtype10 | | ☐ |
| paramtype11 | | ☐ |
| paramtype12 | | ☐ |
| paramtype13 | | ☐ |
| paramtype14 | | ☐ |
| paramtype15 | | ☐ |
| paramtype16 | | ☐ |

param2 is set to the customer Id. param3 is set to the customer account Id. param4 is set to the new account status. These values are used in the stored procedure to update the customer account status.

| Name | Value | Use XPATH? |
|---|---|---|
| paramtype17 | | ☐ |
| paramtype18 | | ☐ |
| paramtype19 | | ☐ |
| paramtype2 | Integer | ☐ |
| paramtype20 | | ☐ |
| paramtype3 | Integer | ☐ |
| paramtype4 | String | ☐ |
| paramtype5 | | ☐ |
| paramtype6 | | ☐ |
| paramtype7 | | ☐ |
| paramtype8 | | ☐ |
| paramtype9 | | ☐ |
| pool | oraclePool | ☐ |
| query_type | PROCEDURE | ☐ |
| result_name | RootResultXmlElement | ☐ |
| row_name | ResultsFromSQL | ☐ |
| sql | (call P_Update_Account(?,?,?,?)) | ☐ |
| StartNewWorkFlow | NO | ☐ |

The PROCEDURE query type invokes a stored procedure in this example.

sql is set to the stored procedure to invoke, where "?,?,?,?" is replaced with the values defined for param1, param2, param3, and param4.

The following example illustrates the same business process using BPML:

```
<process name="ExampleUpdateAccountStatus">
   <sequence>
        <assign to="CustomerId" from="DocToDOM(PrimaryDocument)/Account/Customer/@Id"
append="true"/>
        <assign to="Account" from="DocToDOM(PrimaryDocument)/Account/@Id" append="true"/>
        <assign to="AccountStatus" from="DocToDOM(PrimaryDocument)/Account/Status/text()"
append="true"/>
        <operation name="Lightweight JDBC Adapter">
                <participant name="ExampleLWJDBCBusinessProcess"/>
                <output message="LightweightJDBCAdapterTypeInputMessage">
                <assign to="param1">1</assign>
                        <assign to="paramtype1">Integer</assign>
                        <assign to="param2" from="number(/ProcessData/CustomerId/@Id)"/>
                        <assign to="paramtype2">Integer</assign>
                        <assign to="param3" from="number(/ProcessData/Account/@Id)"/>
                        <assign to="paramtype3">Integer</assign>
                        <assign to="param4" from="/ProcessData/AccountStatus/text()"/>
                        <assign to="paramtype4">String</assign>
                        <assign to="pool">oraclePool</assign>
                        <assign to="query_type">PROCEDURE</assign>
                        <assign to="result_name">RootResultXmlElement</assign>
                        <assign to="row_name">ResultsFromSQL</assign>
                        <assign to="sql">{call P_Update_Account(?,?,?,?)}</assign>
                        <assign to="." from="*"/>
                </output>
                <input message="inmsg">
                        <assign to="." from="."/>
                </input>
        </operation>
   </sequence>
</process>
```

Specify PROCEDURE as the query type when using BPML.

Notice the format used to call stored procedures.

## Calling Oracle Functions

The following example shows a database script for creating a simple Oracle function. The function retrieves and returns the number of products in the database:

```
CREATE OR REPLACE FUNCTION product_count
RETURN number
 IS
   V_product_count number;
BEGIN
   SELECT count(*) INTO v_product_count
   FROM product;
RETURN v_product_count;
END;
/
```

The following example using the GPM illustrates a business process that uses the Lightweight JDBC adapter to start the function:

ExampleGetProductCount.bp

start    Sequence Start    Lightweight JDBC Adapter    Sequence End    End

Service Editor-Lightweight JDBC Adapter

Name    Lightweight JDBC Adapter
Config.  ExampleLWJDBCBusinessProcess

Message To Service | Message From Service

Output Msg    Obtain Message first, then Process Data
Message Name  LightweightJDBCAdapterTypeInputMessage

No input parameters are specified for the function. However, you must specify a parameter type that matches the format of the value returned from the function. In this case, we set paramtype1 to Integer because the function returns the number of products.

| Name | Value | Use XPATH? |
|---|---|---|
| paramtype1 | Integer | ☐ |
| paramtype10 | | ☐ |
| paramtype11 | | ☐ |
| paramtype12 | | ☐ |
| paramtype13 | | ☐ |
| paramtype14 | | ☐ |
| paramtype15 | | ☐ |
| paramtype16 | | ☐ |
| paramtype17 | | ☐ |
| paramtype18 | | ☐ |
| paramtype19 | | ☐ |

Advanced

sql is set to the function to invoke, where "?" is replaced with the value returned from the function and mapped to paramtype1.

| | | |
|---|---|---|
| paramtype19 | | ☐ |
| paramtype2 | | ☐ |
| paramtype20 | | ☐ |
| paramtype3 | | ☐ |
| paramtype4 | | ☐ |
| paramtype5 | | ☐ |
| paramtype6 | | ☐ |
| paramtype7 | | ☐ |
| paramtype8 | | ☐ |
| paramtype9 | | ☐ |
| pool | oraclePool | ☐ |
| query_type | Stored procedure/function | ☐ |
| result_name | TestDBLookupResult | ☐ |
| row_name | ResultsFromSQL | ☐ |
| sql | {? = call product_count()} | ☐ |
| StartNewWorkFlow | This Lightweight JDBC Adapter will not start a new busine... | ☐ |

The following example illustrates the same business process using BPML:

```
<process name="ExampleGetProductCount">
        <sequence>
                <operation name="Lightweight JDBC Adapter">
                        <participant name="ExampleLWJDBCBusinessProcess"/>
                        <output message="LightweightJDBCAdapterTypeInputMessage">
                                <assign to="paramtype1">Integer</assign>
                                <assign to="pool">oraclePool</assign>
                                <assign to="query_type">PROCEDURE</assign>
                                <assign to="result_name">TestDBLookupResult</assign>
                                <assign to="row_name">ResultsFromSQL</assign>
                                <assign to="sql">{? = call product_count()}</assign>
                                <assign to="." from="*"/>
                        </output>
                        <input message="inmsg">
                                <assign to="." from="."/>
                        </input>
                </operation>
        </sequence>
</process>
```

Specify PROCEDURE as the query type when using BPML.

Notice the format used to call functions.

# Chapter 129. Lock Service

The Lock service enables a business process to request, renew, or delete a lock for a particular resource. The following table provides an overview of the Lock service:

| System Name | LockService |
|---|---|
| Graphical Process Modeler (GPM) category | All Services |
| Description | Enables a business process to request, renew, or delete a lock for a particular resource. |
| Business usage | If you have a business process, or resources within a business process, that should never have more than one instance running at a time, you can use the Lock service to prevent other instances of the business process, or just a certain part of the business process, from running until the lock is released. |
| Usage example | You have a business process that uses a configuration of the File System adapter called "Inbound Invoices." To prevent this particular configuration of the File System adapter from being invoked by another business process while processing data in the current business process, you add the Lock service before the File System adapter in the business process, to lock that resource. You add a second instance of the Lock service after the File System adapter in the business process to release the lock once the File System adapter processing is complete. |
| Preconfigured? | Yes |
| Requires third party files? | No |
| Platform availability | All supported Sterling Integrator platforms |
| Related services | None |
| Application requirements | None |
| Initiates business processes? | No |
| Invocation | As part of a business process. |
| Business process context considerations | None |
| Returned status values | Returned status values:<br>• Success<br>• Error |
| Restrictions | None |
| Testing considerations | Use the *Business Process Example* as a test. |
| Supports Transactional Processing? | Yes |

## How the Lock Service Works

The Lock service secures a business process and prevents other business processes from using the locked resources until the lock is released. The Lock service uses the lock key and duration time that you set in the GPM for the business process to identify the lock to set and how long to keep the resources locked. You can lock all of the activities and services used in a business process by adding the Lock service to the beginning of a business process, after the Start and Sequence Start activities.

Or, you can lock just some of the activities or services in a business process by adding the Lock service directly before the activities or services to be locked.

To release the lock, add another instance of the Lock service to the business process directly after the group of locked activities and services. If the entire business process was locked, add the second Lock service to the end of the business process, directly before the End Sequence and End activities.

## Implementing the Lock Service

To implement the Lock service, complete the following tasks:

1. Create a configuration of the Lock service, or use the configuration installed with the Sterling Integrator, LockService. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Lock Service*.
2. Include two instances of the Lock service in your business process (one to lock resources, another to unlock them).
3. Specify field settings for each instance in the GPM as necessary. Ensure that you set the first instance to use the Lock action, and the second to use the Unlock action. Also ensure that you specify the same lock key for both.

## Configuring the Lock Service

Use the field definitions in the following table to set up the service configuration in the GPM:

| Field | Description |
|-------|-------------|
| Config | Select the name of the service configuration from the list. |
| ACTION | Action performed for the requested lock. Valid values are: |
| | • Check – Looks for the existence of a lock that is identified in the LOCK_KEY field. If the lock exists, the LOCK_EXIST output workflow parameter is set to true. If the lock does not exist, LOCK_EXIST is set to false. |
| | • List Locks - Returns a document containing an XML list of details of the current active locks. |
| | • Lock – Create (default). |
| | • Touch – Renew. |
| | • Unlock – Delete. |
| | Optional. **Note:** As a best practice, always use the Lock service in pairs in your business processes—one instance to lock the business process, and one to unlock the business process after the necessary operations have completed. |
| DURATION | Time, in milliseconds, that the lock is applied. Required. The lock will time out or expire after this time. **Note:** If a business process halts due to an error, the service configuration will remain locked until you manually release the lock or restart the Sterling Integrator. |

| Field | Description |
|---|---|
| LOCK_KEY | The key for obtaining the lock. Required. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both. |
| USER | User name associated with the lock (informational only). Optional. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both. |

## Parameters That Must be Added in BPML

The following additional parameters can be used with Lock service by editing the BPML:

| Parameter | Description |
|---|---|
| CLEAR_ON_START_UP | Clears the lock after you restart the Sterling Integrator. Use when you want to ensure that the lock is always clean when the Sterling Integrator is restarted. Optional. Valid values are true and false. |
| IN_TRANSACTION | Determines whether Lock service can participate in multi-step transactions. Optional. Default is false. Valid values are: True - Forces Lock service to participate in multi-step transactions. False - Lock service does not participate in multi-step transactions. Data base will not roll back. In cases where changes are immediately visible to other business processes, this should be false. |

## Business Process Example

The following example illustrates how the Lock service could be used in a business process.

In this business process, a lock is applied by the Lock service. The lock is set to a duration of 600,000 milliseconds, and uses "Lock1" as the lock key. In this business process, the resource being locked is an instance of the Sleep service. Note that the sleep interval (duration) is 45 seconds. The Sleep service is followed by a second instance of the Lock service that releases the lock, thereby freeing up this configuration of the Sleep service configuration for other processes.

**Note:** The business process also includes onFault activities and messages in the event that one of the lock activities fails.

```
<process name="LockExample">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="DURATION">600000</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="CLEAR_ON_START_UP">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
```

```
<sequence name="Process">
  <operation name="Sleep">
    <participant name="TestSleepService"/>
    <output message="Xout">
      <assign to="SLEEP_INTERVAL">45</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="Xin">
      <assign to="." from="*"></assign>
    </input>
  </operation>
  <sequence name="UnLock">
    <operation name="UnLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="ACTION">unlock</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <onFault>
      <assign to="UnLock_Msg" append="true">Failed to obtain an
          unlock!</assign>
    </onFault>
  </sequence>
  <onFault>
    <operation>
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="ACTION">unlock</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
      </input>
    </operation>
  </onFault>
</sequence>
<onFault>
  <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
</onFault>
</sequence>
</process>
```

The following GPM screen shows the example business process graphically. Note that the instance of the Lock service called SetLock is selected and its properties are displayed in the Service Editor in the lower half of the screen:
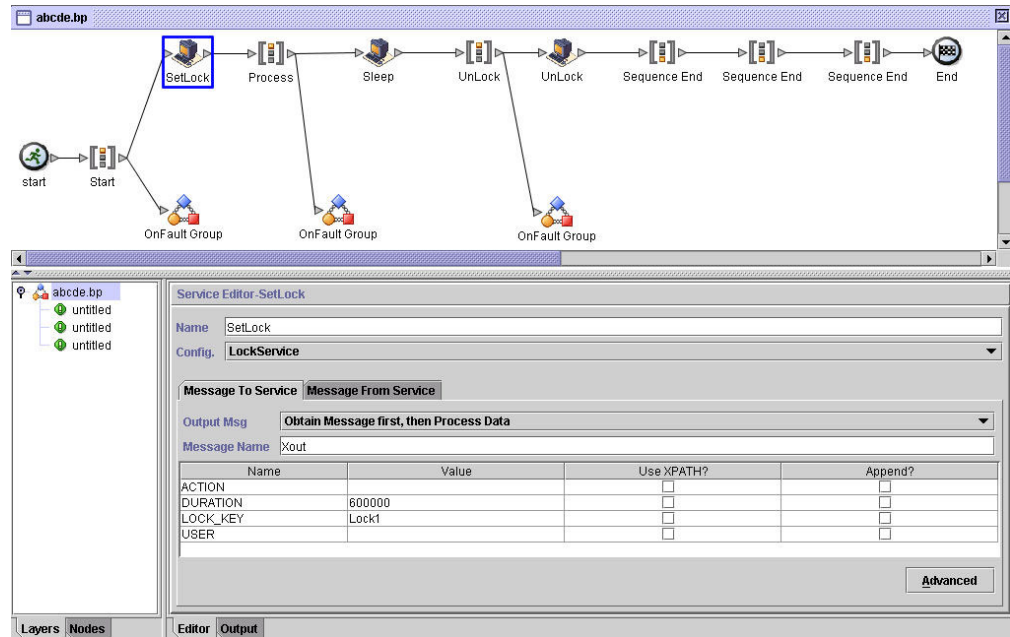
## Lock Service – Frequently Asked Questions

### How do I determine what the duration of a lock should be?

Set the lock duration value carefully (generally 2-3 times what the estimated time of execution will be) so that lock does not time out before the business process reaches the unlock step.

### If the lock or unlock step fails (or any step in the business process between the two Lock services), are there any "cleanup" activities that need to be done?

Halted business processes can be terminated, or they can be left in halted state until the issue is resolved, and then restarted or resumed (as appropriate for the persistence level). The path to take depends on the needs of your business with regard to the business process itself. For example, is it mission-critical? Do other processes depend on its completion? Is the locked resource going to cause other business processes that use it to halt? Was the error caused by a problem in the configuration of the locked resource? Will this need to be corrected before using it again?

There are two general steps to follow first:

1. Check the lock manager page to see what's locked.
2. Check **Business Process** > **BP Monitor** > **Current Processes** for more information about the error.

Then, once you have determined what caused the error, you can decide when to release the lock and when to terminate, restart, or resume the business process.

To manually release a locked resource:

Go to **Operations** > **Lock Manager**, and click **Go!** in the List panel. The locked resources are displayed on a results page. Locate the resources from your business process that are locked and clear the Lock checkbox for the resources.

To terminate, restart, or resume a business process:

Go to **Business Process** > **Monitor** > **Current Processes** and select the ID of the halted instance of your business process. From the page displayed, you can select the appropriate action for this business process: terminate, restart, or resume.

## Are there any best practices for using the Lock service in a business process?

Use the Lock service twice in a business process—one to lock resources and one to unlock them. The first instance precedes the resources to be locked and the second instance follows them. See the *Business Process Example* for a graphical representation. Do not use just one instance of the Lock service in a business process and let it expire instead of using a second Lock service to release the lock.

# Index

## A

absolute path:definition 424
Action query:example 849
adapter:B2B Mail Client 27
adapter:B2B SMTP Client 45
adapter:BEA Tuxedo 59
adapter:BP Fault Log 73
adapter:BP Fault XML Log 77
adapter:Command Line 223
adapter:Command Line 2 239
adapter:Connect\:Direct Requester 91
adapter:Connect\:Enterprise UNIX
  Server 185
adapter:description 223
adapter:E-5 2000 341
adapter:E5 Client
  Configuration;adapter:E5 Server
  Configuration 341
adapter:E5 Obtain 341
adapter:EJB 391
adapter:FTP Client 445
adapter:FTP Server 529, 539
adapter:Gentran\:Server Windows 561
adapter:GXS ICS FTP 567
adapter:HTTP Client 617
adapter:HTTP Server 665
adapter:IBM IE FTP 671
adapter:IM 677
adapter:Instant Messaging (IM) 677
adapter:iWay 707
adapter:Java Database Connectivity
  (JDBC) 717
adapter:JMS 1.1 Async Receive 771
adapter:JMS Queue 741
adapter:JMS Topic 751
adapter:LDAP 815
adapter:Lightweight JDBC 823
adapters used with Perimeter
  servers:Connect\:Direct Requester
  adapter 91
adapters used with Perimeter
  servers:Connect\:Enterprise UNIX
  Server adapter 185
adapters used with Perimeter servers:FTP
  Client adapter 445
adapters used with Perimeter servers:FTP
  Server adapter 529, 539
adapters used with Perimeter
  servers:HTTP Server adapter 665
adapters used with Perimeter servers:IBM
  IE FTP adapter 671
AFT Route Progress Event Reporting
  service:business process example 2
AFT Route Progress Event Reporting
  service:configuring 2
AFT Route Progress Event Reporting
  service:how it works 1
AFT Route Progress Event Reporting
  service:implementing 1
AFT Route Progress Event Reporting
  service:overview 1

AFT Route Progress Event Reporting
  service:parameters passed from BP to
  service 2
AFT Route Progress Event Reporting
  service:parameters passed from service
  to BP 2
Alert service:Alert Notification business
  process 5
Alert service:alert.properties file default
  configuration 7
Alert service:alerters 5
Alert service:configuring additional
  alerters 7
Alert service:description 3
Alert service:filters 5
Alert service:how it works 3
Alert service:implementing new alerter
  classes 9
Alert service:implementing new filter
  classes 8
Alert service:main alerter 4
Alert service:modifying predefined
  business processes 6
Alert service:overview 3
alerter 5
Application adapters:BEA Tuxedo
  adapters 59, 60
Application adapters:Instant Messaging
  Adapter Suite 677
Archive Business Process service:how it
  works 11
Archive Business Process
  service:overview 11
Attachment Parsing service:Business
  scenario example 14
Attachment Parsing service:output from
  business process to service 15
Attachment Parsing service:overview 13
Auto Terminate service:business process
  examples 21
Auto Terminate service:configuring 18
Auto Terminate service:how it works 17
Auto Terminate service:implementing 18
Auto Terminate service:overview 17

## B

B2B Lookup System service:overview 25
B2B Mail Client adapter:configuring 29
B2B Mail Client adapter:data flow 27, 33
B2B Mail Client adapter:how it
  works 28
B2B Mail Client
  adapter:implementing 29
B2B Mail Client adapter:overview 27
B2B Send BPML example 39
B2B Send service:BPML example 42
B2B Send service:business scenario 37
B2B Send service:how it works 36
B2B Send service:implementing 40
B2B Send service:usage example 42

B2B Send System service 35
B2B services, about 36
B2B SMTP Client adapter 45
Backup service:configuring 49
Backup service:how it works 47
Backup service:implementing 48
Backup service:overview 47
Batch Processor service 51
Batch Processor service:configuring 52
Batch Processor service:implementing 52
BEA Jolt 59, 60
BEA Tuxedo adapters 59
BEA Tuxedo adapters:Application
  Services adapter 61
BEA Tuxedo adapters:configuring 62
BEA Tuxedo adapters:how they work 60
BEA Tuxedo adapters:implementing 60,
  61
BEA Tuxedo adapters:installing 62
BEA Tuxedo adapters:notifications 61
BEA Tuxedo adapters:retrieving
  services 61
BEA Tuxedo adapters:Subscribe
  adapter 61
Begin Transaction service:how it
  works 68, 402
Begin Transaction
  service:implementing 70, 404
BP Exception service:configuring 72
BP Exception service:how it works 71
BP Exception service:implementing 71
BP Fault Log adapter:business process
  error message logs 74
BP Fault Log adapter:configuring 74
BP Fault Log adapter:failing a business
  process 74
BP Fault Log adapter:how it works 74
BP Fault Log adapter:including body of
  data with message 74
BP Fault Log adapter:invalid data 74
BP Fault Log adapter:log levels 74
BP Fault Log adapter:overview 73
BP Fault Log adapter:troubleshooting
  business processes 74
BP Fault XML Log
  adapter:configuring 78
BP Fault XML Log
  adapter:implementing 77
BP Fault XML Log adapter:overview 77
BP Metadata Info service:BP example 83
BP Metadata Info service:configuring 82
BP Metadata Info
  service:implementing 82
BP Metadata Info service:output from BP
  to service 83
BP Metadata Info service:output from
  service to BP 82
BP Metadata Info service:overview 81
BP Report service: business process
  example 86

# Notices

This information was developed for products and services offered in the U.S.A.

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing*

*IBM Corporation*

*North Castle Drive*

*Armonk, NY 10504-1785*

*U.S.A.*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*

*Legal and Intellectual Property Law*

*IBM Japan Ltd.*

*19-21, Nihonbashi-Hakozakicho, Chuo-ku*

*Tokyo 103-8510, Japan*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be

incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation*

*J46A/G4*

*555 Bailey Avenue*

*San Jose, CA 95141-1003*

*U.S.A.*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© IBM 2015. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2015.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

**IBM** ®

Product Number:

Printed in USA