Sterling Integrator

Services and Adapters Build Updates

Version 5.0

Sterling Integrator

Services and Adapters Build Updates

Version 5.0

Note
Before using this information and the product it supports, read the information in "Notices" on page 173.

Contents

Chapter 1. Introduction to Build Updates

This document provides information about services and adapters provided in IBM® Sterling Integrator Version 5.0. Documentation limited to specific build numbers is identified in the topic title.

Chapter 2. Build 5007 or Higher

FTP Server Adapter

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 put files into, or get files from, a mailbox.
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 disabled by default. You can enable the preconfigured FTP Server adapter or create a new configuration.
Requires third party files?	Certicom SSL Library provided
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). To access a mailbox, you must have permission to that mailbox and all mailboxes between it and your virtual root. If a user exceeds the maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	The FTP Server adapter does not directly initiate business processes. However, mailbox activities can trigger routing rules.
Invocation	Not used in business processes
Business process context considerations	None
Returned status values	None

Restrictions	Restrictions: • FTP Server is tightly integrated with the application'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). Basic functions are supported to integrate with the 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. 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 **`MailboxLoginWithoutVirtual RootPermission`**. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes.
Persistence level	None. This adapter does not have a pre-set persistence level.

Testing considerations	At application 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 installed configuration 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:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration. Required.
Select a Group	Not applicable for this adapter. Do not change default value.
FTP Server Listen Port	Port number that the FTP Server should bind to and listen on for connection requests. The default value depends on your system platform and on configuration. Required.
Active Data Port Range	Range of ports the server can allocate for the transfer of data to or from the FTP client in active mode. Optional. Example values are: • 1024-2048 • 2222 • 3000-4000 **Note:** You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799If left blank, the server selects available system ports.

Field	Description
Passive Data Port Range	Range of ports the server can allocate for the transfer of data to or from the FTP client in passive mode. Optional. Example values are: • 1024-2048 • 2222 • 3000-4000 **Note:** You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799If left blank, the server will choose available system ports.
Perimeter Server	Select a perimeter server from the list. Default is node1 and 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 another process binds 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	Tuning parameter indicating 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:** Do not change the default value unless instructed otherwise by Sterling Commerce support.
Maximum Number of Threads	Tuning parameter indicating 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:** Do not change the default value unless instructed otherwise by Sterling Commerce support.
NAT Address	Specifies the NAT IP address the FTP server should send to the user FTP client in 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 time. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999.

Field	Description
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.
Document Storage	Indicates whether the body of the request document must be stored on the file system or in the database. Valid values are: • System Default – If your system administrator has changed the default value, this ensures the correct location is used. • Database – Body of the request document will be stored in the database. • File System (default) – This is the default value, but it can be changed. Contact your system administrator to see if the default has been changed. Required. **Note:** For more information about document storage types, see *Managing Services and Adapters*.
Should the adapter be restricted to a certain group of users?	Select Yes or No to indicate whether to restrict access to 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 range of ports to 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?	Places the user, upon logging in, into a directory (mailbox) that corresponds to their user ID. Valid values are: • Yes – Upon login, the user is automatically placed in a directory that matches their 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 their user ID. **Caution**: Do not select Yes if any user IDs differ only by case (example: jsmith and JSmith). Unlike user IDs, mailbox names are not case-sensitive. • No – User is placed in the virtual root directory.
Users	Select a list of users who are granted permission to access the server.

Field	Description
Groups	Select a list of groups who are granted permission to access the server.
User Active Ports	Any port number or a range of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or a range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning.Optional. Examples of valid values are: • 3000 • 4000-5000, 6000
User Passive Ports	Any port number or a range of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or a range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Optional. Examples of valid values are: • 3000 • 4000-5000, 6000
Group Active Ports	Any port number or a range of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or a range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Optional. Examples of valid values are: • 3000 • 4000-5000, 6000
Group Passive Ports	Any port number or a range of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Optional. Examples of valid values are: • 3000 • 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.

Field	Description
Extractable	Whether the message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Valid values are Yes and No. Default is Yes. Optional.
SSL	Whether Secure Sockets Layer (SSL) is active. Required. Valid values are: • None – If SSL is requested by a client it will be rejected (default) • Optional – SSL is used if requested by a client • Must – Clients that do not request SSL are not allowed to authenticate **Note:** If Optional or Must is selected, 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: • ALL • WEAK – Often required for international e-commerce, because government regulations prohibit STRONG encryption from being exported • 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.

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 • PASS – Password • CWD – Change Working Directory • CDUP – Change to Parent Directory • QUIT – Logout
Transfer Parameter Commands	• PORT – Data port • PASV – Passive mode • TYPE – Representation type (ASCII and Binary) • STRU – File Structure (File) • MODE – Transfer mode (Stream)
Service Commands	• DELE – Delete • RETR – Retrieve • STOR – Store • ABOR – Abort • PWD – Print Working Directory • XPWD – Print Working Directory (legacy format) • LIST – List • NLST – Name List • HELP – Help • NOOP – No Operation • RNFR – Rename From • RNTO – Rename To • SITE – Site Parameter (CPWD and HELP) • SYST – System • MDTM – Last-modified time of a given file on a remote host • SIZE – Return size of a remote file
Security Commands	• AUTH – Authentication/Security Mechanism • CCC – Clear Command Channel • PBSZ – Protect Buffer Size • PROT – Data Channel Protection Level • REST – Restart

FTP Server Functions Not Supported

The following table lists the FTP functions that are not supported with the FTP Server adapter:

Category	Commands Not Supported
Access Control commands	• ACCT – Account • SMNT – Structure Mount • REIN – Re-initialize

Category	Commands Not Supported
Transfer Parameter Commands	• TYPE – Representation type (EBCDIC and Local Byte) • STRU – File Structure (Record and Page) • MODE – Transfer mode (Block and Compressed)
Service Commands	• STOU – Store Unique • APPE – Append • ALLO – Allocate • RMD – Remove Directory • MKD – Make Directory • STAT – Status

Activity Types for the FTP Server Adapter

This adapter reports the following activities to the Services Controller for activity monitoring:

- PUT – Adds a file to a mailbox
- GET – Retrieves a file from a mailbox
- Session – Records all activity after connection

User Exits

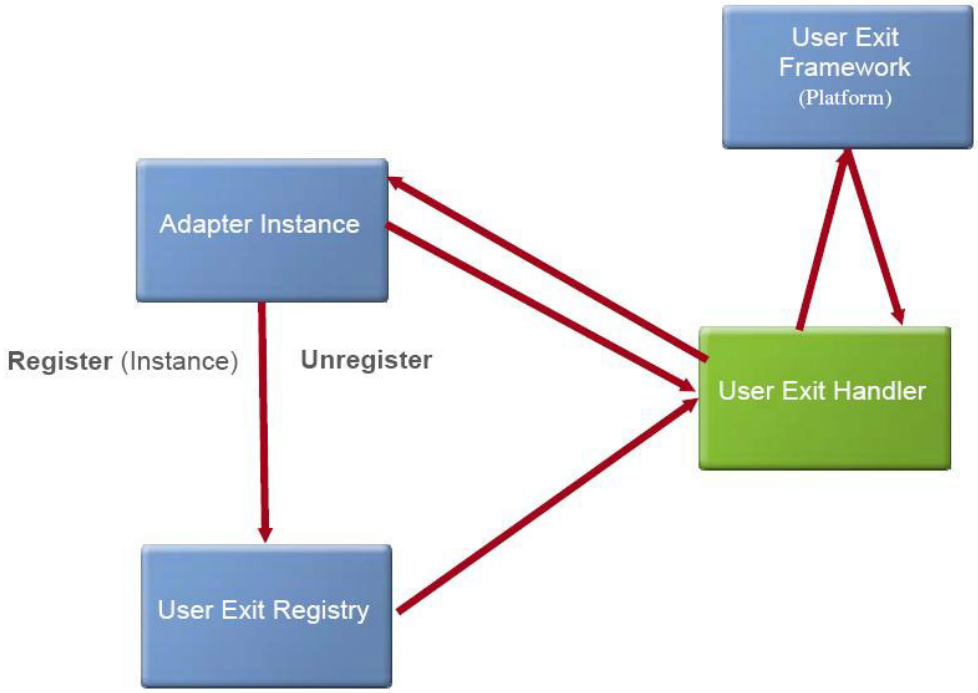
User exits are a set of predefined points that allow you to implement customized functions by adding custom code to perform a desired operation, thereby extending the functionality of the adapter.

The User Exit framework consists of the following components, plus a handler that interacts with all the components to perform the desired task:

- Adapter or service that needs to use the execution framework
- User exit registry that can be queried for all user exits configured for a particular adapter instance. All user exits are registered and maintained in this registry.
- User exit execution framework that allows you to obtain the references to the user exit implementation and to execute the user exit

The handler's reference is maintained by the adapter, which refers to the registry and the execution framework.

The following figure shows the user exit architecture:



The following table provides the generic properties that are defined for a user exit:

Property	Description
Implementations	Contains the list of custom code implementation classes that must be invoked when a user exit is executed. The implementation classes are invoked in the order they were added. If a user exit implementation fails, the next user exit implementation in the chain is not invoked.
return.on.exception	Determines the result if an exception occurs when a user exit is being executed. You should set the value to false only if the user exit is critical and displays a failure.
pool.size (integer value 1 - 10)	These properties are used to manage thread pools for executing the user exits.
maximum.queue.length (integer value 1 – 100)	
wait.time (integer value 1 - 600) in seconds	

Configuring User Exits

The following user exit points are defined in the FTP Server Adapter:

- `com.sterlingcommerce.woodstock.userexit.services.ftpserver.interfaces.IFtpServerUserExit_OnFileReceiveBeforeCommit`
- `com.sterlingcommerce.woodstock.userexit.services.ftpserver.interfaces.IFtpServerUserExit_OnCwdCommandBeforeExecute`
- `com.sterlingcommerce.woodstock.userexit.services.ftpserver.interfaces.IFtpServerUserExit_OnUnknownSiteSubCommand`

The interfaces are provided through separate jar files present in the `install_dir/install/userexit/jars` (`install_dir\install\userexit\jars` for Windows) directory.

Perform the following tasks to configure user exit points:

1. Write the code to implement the interface for the desired point.
2. Add the custom code classes to a .jar file.
3. Add the path of the .jar file to the dynamicclasspath.cfg file in the `install_dir/install/properties` (`install_dir\install\properties` for Windows) directory.
4. Restart Sterling Integrator.
5. Navigate to the `install_dir/properties/userexit` (`install_dir\properties\userexit` for Windows) directory and locate FtpServerUserExits.xml file.
6. Edit FtpServerUserExits.xml file and add an entry for each implementation as shown. The user exits are executed in the same order as they appear.

```
<bean id="com.sterlingcommerce.woodstock.userexit.services.ftpserver.interfaces.
IFtpServerUserExit_OnCwdCommandBeforeExecute" class="com.sterlingcommerce.
woodstock.userexit.services.ftpserver.FtpServerUserExit">
    <property name="implementations">
     <list>
   <value>implementation1</value>
   <value> implementation2</value>
     </list>
    </property>
    <property name="generalParameters">
   <props>
    <prop key="return.on.exception">false</prop>
    <prop key="pool.size">5</prop>
    <prop key="maximum.queue.length">5</prop>
    <prop key="wait.time">10</prop>
    <prop key="execution.threshold.time">600000</prop>
   </props>
    </property>
    </bean>
```

   Remove all values to deactivate the user exit points.
7. Restart the FTP Server adapter instance to apply the changes.

   Note: Restart only the adapter instance if you modify implementation class list and other properties.

SFTP Client GET Service

The following table provides an overview of the SFTP Client GET service:

System name	SFTP Client GET Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	This service is used to retrieve one or more documents from a specified directory on the trading partner's SFTP server.
Business usage	Use this service to retrieve one or more documents from a trading partner and move them into Sterling Integrator when the SFTP protocol is required as the transport mechanism.

Usage example	A business process is executed to retrieve a specified file or files from the external trading partner. Sterling Integrator uses the SFTP Client GET service, working through the SFTP Client adapter, to retrieve a file or files from a specified directory on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Sterling Integrator supported platforms.
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: • SFTP Client Begin Session service • SFTP Client CD service • SFTP Client DELETE service • SFTP Client End Session service • SFTP Client GET service • SFTP Client LIST service • SFTP Client MOVE service • SFTP Client PUT service • SFTP Client PWD service
Application requirements	An SFTP Server at the external trading partner location
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0 – Success, 1 – Error
Restrictions	None
Persistence level	System default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see the *SFTP Client adapter* documentation. For further information, go to Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client GET service:

Field	Description
RemoteFileName	Name of the file to be retrieved from the remote trading partner. Optional. You cannot use this parameter if RemoteFilePattern is specified. **Note:** Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank.
ResponseTimeout	Maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is the ResponseTimeout value specified in the SFTP Client Begin Session service.
SessionToken	Returned SessionToken from the Begin Session service. Required.
RemoteFilePattern	File filter pattern. Using this field activates multiple-get mode. Optional. You cannot use this parameter if RemoteFileName is specified. **Note:** Either RemoteFileName or RemoteFilePattern must be specified. Both cannot be left blank.
RetrieveErrorSetSuccess	SFTP Client Get service will succeed in case of any error when RetrieveErrorSetSuccess field is set to `YES`. Optional. Valid values are `YES` and `NO`.

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client GET service to the business process:

Parameter	Description
ServiceStartTime	Date/time stamp for when the service started
DocumentId	Provides information about the file retrieved as a result of the GET service.
ServerResponse	SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: • 0 - OK • 1 - End of File • 2 - No Such File • 3 - Permission Denied • 4 - General Failure • 5 - Bad Message • 6 - No Connection • 7 - Connection Lost • 8 - Operation Unsupported
ServiceEndTime	Date/time stamp for when the service ended

Parameter	Description
Primary Document	File retrieved as a result of the GET service

Business Process Example

The following business process excerpts illustrate using the SFTP Client GET service:

- Process to get a binary file named FileNameToGet from the server

```
[[Insert begin session here]]

<operation name="SFTP Client GET Service">
      <participant name="SFTPClientGet"/>
      <output message="SFTPClientGetServiceTypeInputMessage">
 <assign to="RemoteFileName" >FilenameToGet</assign>
       <assign to="SessionToken" from="SFTPClientBeginSessionServiceResults/
SessionToken/text()"></assign>
     </output>
      <input message="inmsg">
       <assign to="SFTPClientGetServiceResults" from="*"></assign>
     </input>
    </operation>
[[Insert end session here]]
```

- Process using a multiple get command

```
[[Insert begin session here]]
<operation name="SFTP Client Multiple GET Service">
<participant name="SFTPClientGet"/>
<output message="SFTPClientGetServiceTypeInputMessage">
<assign to="RemoteFilePattern">*.txt</assign>
<assign to="SessionToken"
from="SFTPClientBeginSessionServiceResults/SessionToken/text()"></assign>
</output>
<input message="inmsg">
<assign to="SFTPClientGetServiceResults" from="*"></assign>
</input>
</operation>
[[Insert end session here]]
```

Correlations and Document Tracking

The following table details the correlations available from the SFTP Client GET service for document tracking:

Key	Values
ACTION	Get, Put
Direction	Inbound, Outbound
Protocol	SFTP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost
Username	username
RemoteFile	filename

SFTP Client PUT Service

The following table provides an overview of the SFTP Client PUT service:

System name	SFTP Client PUT Service
Graphical Process Modeler (GPM) category	All Services, B2B Protocols > SFTP Client
Description	Used to place a document or documents in a specified directory on the trading partner's SFTP server.
Business usage	Use this service to transfer a document or documents from Sterling Integrator to a trading partner when the SFTP protocol is required as the transport mechanism.
Usage example	A business process is executed that translates a document or documents to send to a trading partner. After the translation, Sterling Integrator uses the SFTP Client PUT service, working through the SFTP Client adapter, to place the document or documents in a specified directory on the trading partner system.
Preconfigured?	Yes. To implement, use the preconfigured service in a business process.
Requires third party files?	No
Platform availability	All Sterling Integrator supported platforms.
Related services	The following services are related. Configured in a business process, they initiate the SFTP Client adapter to perform their operations: • SFTP Client Begin Session service • SFTP Client CD service • SFTP Client DELETE service • SFTP Client End Session service • SFTP Client GET service • SFTP Client LIST service • SFTP Client MOVE service • SFTP Client PUT service • SFTP Client PWD service The SFTP Client PUT service must be placed between an SFTP Begin Session service and an SFTP End Session service. It may be used to put a document that is returned from an SFTP Client GET service.
Application requirements	An SFTP Server at the external trading partner location
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	None
Returned status values	0 – Success, 1 – Error

Restrictions	None
Persistence level	System default
Testing considerations	To test this service, run the SFTPClientDemoAllServices business process and verify that it completes successfully. For more information about the SFTPClientDemoAllServices business process, see the *SFTP Client adapter* documentation. For debugging information, go to Operations > System > Logs > SFTP Client Adapter and Services

Input from Business Process to Service

The following table contains the parameters passed from the business process to the SFTP Client PUT service:

Field	Description
DocumentId	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 SFTP 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.
RemoteFileName	Name of the file used to place the document on the remote trading partner server. If not specified, the name of the document will be used. Optional.
ResponseTimeout	Maximum number of seconds it can take for the trading partner system to respond before the session times out and terminates. If a number less than 30 is specified, 30 seconds will be used. Optional. Default is the ResponseTimeout value specified in the SFTP Client Begin Session service.
SessionToken	Returned SessionToken from the Begin Session service. Required.
Primary Document	File transferred as a result of the PUT service.

DocumentList	List of documents to PUT to the remote server. Each item must be a DocumentId. A list could look like the following example:
	`<DocumentList>`
	`  <DocumentId>12345</DocumentId>`
	`  <DocumentId>67890</DocumentId>`
	`</DocumentList>`

Field	Description
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: • Yes – Use document body name • No – (Default) Use document name

Output from Service to Business Process

The following table contains the parameters passed from the SFTP Client PUT service to the business process:

Parameter	Description
ServerResponse	The SFTP server response, which may include a reply code and any text associated with the reply code. Valid values are: • 0 - OK • 2 - No Such File • 3 - Permission Denied • 4 - General Failure • 5 - Bad Message • 6 - No Connection • 7 - Connection Lost • 8 - Operation Unsupported

Business Process Example

The following business process excerpt uses the SFTP Client Adapter to send the primary document from Sterling Integrator to the remote SFTP server using the SFTP Client PUT service:

```
[[Insert Begin Session]]

    <operation name="SFTP PUT SERVICE">
    <participant name="SFTPClientPut"/>
      <output message="PutRequest">
       <assign to="SessionToken"
         from="/ProcessData/SftpBeginSessionServiceResults/SessionToken/text()">
       </assign>
       <assign to="RemoteFileName">FilenameToPut</assign>
       <assign to="." From="PrimaryDocument"></assign>
     </output>
      <input message="inmsg">
       <assign to="SftpPutServiceResults" from="*"></assign>
     </input>
    </operation>
[[Insert End Session]]
```

The following business process excerpt uses the SFTP Client Adapter to send a document received from a GET from Sterling Integrator to the remote SFTP server:

```
[[Insert Begin Session]]
   <operation name="Get">
      <participant name="SFTPClientGet"/>
```

```
      <output message="GetRequest">
       <assign to="SessionToken"
             from="/ProcessData/SftpBeginSessionResults/SessionToken/text()">
       </assign>
       <assign to="RemoteFileName">FilenameToGet</assign>
      </output>
       <input message="GetResults">
        <assign to="GetResults" from="DocumentId"/>
      </input>
  </operation>
    <operation name="Put">
      <participant name="SFTPClientPut"/>
     <output message="PutRequest">
       <assign to="SessionToken"
             from="/ProcessData/SftpBeginSessionResults/SessionToken/text()">
       </assign>
       <assign to="." From="/ProcessData/GetResults/DocumentId"/>
    <input message="SFtpPutResults">
       <assign to="PutResults" from="*"></assign>
    </input>
    </operation>
[[Insert End Session]]
```

The following business process uses the SFTP Client adapter to send all documents received from a GET operation from Sterling Integrator to the remote SFTP server:

```
[[Insert Begin Session]]
<operation name="Get">
<participant name="SFTPClientGet"/>
<output message="GetRequest">
<assign to="SessionToken"
from="/ProcessData/SftpBeginSessionResults/SessionToken/text()">
</assign>
<assign to="RemoteFilePattern">*.*</assign>
</output>
<input message="GetResults">
<assign to="GetResults" from="DocumentList"/>
</input>
</operation>
<operation name="Put">
<participant name="SFTPClientPut"/>
<output message="PutRequest">
<assign to="SessionToken"
from="/ProcessData/SftpBeginSessionResults/SessionToken/text()">
</assign>
<assign to="." From="/ProcessData/GetResults/DocumentList"/>
<input message="SFtpPutResults">
<assign to="PutResults" from="*"></assign>
</input>
</operation>
[[Insert End Session]]
```

Correlations and Document Tracking

The following table details the correlations available from the SFTP Client PUT service for document tracking:

Key	Values
ACTION	Get, Put
Direction	Inbound, Outbound
Protocol	SFTP
RemoteHostAddress	remoteAddress
RemoteHostName	remoteHost

Key	Values
Username	username
RemoteFile	filename

Chapter 3. Build 5006 or Higher

Data Sweeper Service

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: • EDIINT • Correlations • Document Clones • Document Life Span • GUID • Performance Engine Stats • Workflow Context • Workflow ID **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: • HP-UX • IBM AIX • IBM iSeries (OS/400) • Microsoft Windows 2000 • RedHat AS • Sun Solaris • United Linux

System name	Data Sweeper Service
Related services	None **Note:** The Data Sweeper service references the `dataSweeper.properties` file in the `<SIInstallDir>/properties` directory.
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 • 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>
```

FTP Server Adapter

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 put files into, or get files from, a mailbox in this application.
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 with this application, but is disabled by default. You can enable the preconfigured FTP Server adapter, or create a new configuration from the application.
Requires third party files?	Certicom SSL Library (currently available in the application)
Platform availability	All supported platforms for this application
Related services	None
Application requirements	To log in to the FTP server, you must have permission to your virtual root (either explicitly assigned or defaulted). To access a mailbox, you must have permission to that mailbox and all mailboxes that may be between it and your virtual root. If a user exceeds a maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	The FTP Server adapter does not directly initiate business processes. However, mailbox activities can trigger routing rules.
Invocation	Not used in business processes
Business process context considerations	None
Returned status values	None

System name	FTP Server Adapter
Restrictions	• FTP Server is tightly integrated with the application'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 application mailbox system such as list message and sub-mailbox, send and extract message to/from mailbox. • FTP Server is not integrated with business process invocation when processing a request from a client. • The home directory for FTP is a virtual root mailbox in the application. Mailboxes include both extractable and nonextractable messages. When accessing a mailbox using the FTP Server adapter, only extractable messages are displayed. To change this default behavior, edit the ftpserver.properties file and set **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 Serveradapter 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 `MailboxLoginWithoutVirtual RootPermission`. With this permission, you can log in and list files in a mailbox, but cannot retrieve or place files. This restricted permission only applies to the virtual root mailbox and does not impact operation on submailboxes.
Persistence level	None. This adapter does not have a pre-set persistence level.

System name	FTP Server Adapter
Testing considerations	At application 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 in the application:

Field	Description
Name	Unique and meaningful name for the adapter configuration. Required.
Description	Meaningful description for the adapter configuration, for reference purposes. Required.
Select a Group	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. Example values are: • 1024-2048 • 2222 • 3000-4000 **Note:** You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799 If left blank, the server will choose available system ports.

Field	Description
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. Example values are: • 1024-2048 • 2222 • 3000-4000 **Note:** You can enter double ranges separated by commas, as shown in this example: 10500-10599,10700-10799 If left blank, the server will choose available system ports.
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.
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.

Field	Description
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.
Document Storage	Indicates whether the body of the request document must be stored on the file system or if it should be in the database. Valid values are: • System Default - If your system administrator has changed the installed default of File System, this ensures that the correct location is used. • Database - Body of the request document will be stored in the database. • File System (default) - This is the default value when the application is installed, but it can be changed. Contact your system administrator to see if the default has been changed. Required. **Note:** For more information about document storage types, see *Managing Services and Adapters*.
Support for concurrent duplicate-named file transfers	Allows sending the files with same name, concurrently to the same Mailbox using the same username. It also allows partners to receive multiple duplicate files with the same name, concurrently, but with different content. Valid values are: • Limited (resume of file transfers supported) - Default. This is the default value. The file transfer can be resumed if transfer fails from the point of failure. You cannot transfer duplicate files with the same name concurrently to the same Mailbox and the same username. • Full, concatenate duplicate-named files on a GET (resume of file transfers not supported) - Supports sending duplicate files with the same name concurrently to the same Mailbox using the same username. The files with the same name are concatenated on a GET operation. You cannot resume broken file transfers. • Full (resume of file transfers not supported) - Supports sending duplicate files with the same name concurrently to the same Mailbox using the same username. The files with the same name are not concatenated on GET operation. The GET operation of a duplicate file would retrieve the latest extractable message from the Mailbox. You cannot resume broken file transfers.

Field	Description
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?	Places the user, upon logging in, into a directory (mailbox in the application) that corresponds to his or her user ID. Valid values are: • 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. **Note:** Do not select Yes if there is any chance that users of your application might have user IDs that differ only by case (example: jsmith and JSmith). Unlike user IDs, mailbox names in this application are not case-sensitive. • No - The user is placed in the virtual root directory.
Users	Select a list of users who are granted permission to access the server.
Groups	Select a list of groups who are granted permission to access the server.
User Active Ports	Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are: • 3000 • 4000-5000, 6000 Optional.

Field	Description
User Passive Ports	Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are: • 3000 • 4000-5000, 6000 Optional.
Group Active Ports	Any port number, range, or ranges of port numbers to be used as ACTIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are: • 3000 • 4000-5000, 6000 Optional.
Group Passive Ports	Any port number, range, or ranges of port numbers to be used as PASSIVE port. Valid values are valid, available port numbers or range of port numbers. Ranges are separated by hyphens. Multiple entries must be separated by commas. Spaces do not affect the meaning. Examples of valid values are: • 3000 • 4000-5000, 6000 Optional.
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. Valid values are Yes and No. Optional. Default is Yes.

Field	Description
SSL	Whether Secure Sockets Layer (SSL) is active. Required. Valid values are: • None - If SSL is requested by a client it will be rejected. (default) • Optional - SSL is used if requested by a client. • Must - Clients that do not request SSL are not allowed to authenticate. **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. Valid values are: • ALL • WEAK - Often required for international e-commerce, because government regulations prohibit STRONG encryption from being exported. • STRONG - Default. Required if SSL option is Must or Optional.
Key Certificate (System Store)	Private key and certificate for server authentication. Used to encrypt and decrypt messages. Required if SSL option is Must or Optional.
CA Certificates	Certificate used to validate the certificate of an FTP client. This is the public key. If no CA certificate is chosen, no client certification is performed. Optional.
Clear Command Channel	Indicates that communication across the command channel is not encrypted after authentication is completed. Optional.

FTP Server Functions Supported

The following table lists the FTP functions that are supported with the FTP Server adapter:

Category	Commands Supported
Access Control Commands	• USER - User name • PASS - Password • CWD - Change Working Directory • CDUP - Change to Parent Directory • QUIT - Logout

Category	Commands Supported
Transfer Parameter Commands	• PORT - Data port • PASV - Passive mode • TYPE - Representation type (ASCII and Binary) • STRU - File Structure (File) • MODE - Transfer mode (Stream)
Service Commands	• DELE - Delete • RETR - Retrieve • STOR - Store • ABOR - Abort • PWD - Print Working Directory • XPWD - Print Working Directory (legacy format) • LIST - List • NLST - Name List • HELP - Help • NOOP - No Operation • RNFR - Rename From • RNTO - Rename To • SITE - Site Parameter (CPWD and HELP) • SYST - System • MDTM - Last-modified time of a given file on a remote host • SIZE - Return size of a remote file
Security Commands	• AUTH - Authentication/Security Mechanism • CCC - Clear Command Channel • PBSZ - Protect Buffer Size • PROT - Data Channel Protection Level • REST - Restart

FTP Server Functions Not Supported

The following table lists the FTP functions that are not supported with the FTP Server adapter:

Category	Commands Not Supported
Access Control Commands	• ACCT - Account • SMNT - Structure Mount • REIN - Re-initialize
Transfer Parameter Commands	• TYPE - Representation type (EBCDIC and Local Byte) • STRU - File Structure (Record and Page) • MODE - Transfer mode (Block and Compressed)

Category	Commands Not Supported
Service Commands	• STOU - Store Unique • APPE - Append • ALLO - Allocate • RMD - Remove Directory • MKD - Make Directory • STAT - Status

Activity Types for the FTP Server Adapter

This adapter reports the following activities to the Services Controller for activity monitoring:

- PUT - Adds a file to a mailbox
- GET - Retrieves a file from a mailbox
- Session - Records all activity after connection

SFTP Server Adapter

The SFTP Server adapter has the following major features:

- Uses perimeter services
- Uses Mailbox subsystem as its repository (virtual roots)
- Routing rules for items placed in Mailbox can be used to trigger a business process
- Supports SSH2 with SFTP version 3 or lower
- Supports inbound SSH/SFTP and SSH/SCP protocols

The following table provides an overview of the SFTP Server adapter:

System name	SFTP Server Adapter
Graphical Process Modeler (GPM) category	None
Description	Receives and processes requests from external trading partners that are submitted through the SFTP protocol or SCP protocol.
Business usage	Use this adapter to enable external SFTP clients or SCP clients to put files into, or get files from, a mailbox in this application.
Usage example	A trading partner uses an SFTP client to retrieve a business document from a mailbox. The SFTP Server adapter receives and processes the trading partner request.
Preconfigured?	DemoAllSFTPServerAdapter is fully preconfigured and enabled when you perform the demo procedure. See *Run SFTPClientDemoAllServices*.
Requires third party files?	No
Platform availability	All supported platforms for this application.
Related services	Perimeter services

System name	SFTP Server Adapter
Application requirements	An SFTP or SCP client at the external trading partner location. When this adapter is configured with a "non-local-mode" perimeter server, the perimeter server must be installed and running. The perimeter server is typically installed in a DMZ environment, separated from the application by a firewall. Refer to the perimeter services documentation for details on installing and running that component. If users exceed a maximum number of failed login attempts, the FTP Server adapter locks the user out. The lock must be reset before the user can access the server again.
Initiates business processes?	No
Invocation	This adapter is not invoked from a business process.
Business process context considerations	None
Returned status values	Not applicable

System name	SFTP Server Adapter
Restrictions	Restricted to platforms that support Java SDK version 1.4 and above. Transfer resumption is disabled by default. To enable transfer resumption and listing documents that are in the staging area, edit the `sftp.properties` file (located at `<install_dir>`/properties/sftp.properties.in) to set listStagedDocuments = True. To support transfer resumption, the SFTP Server Adapter keeps partial documents in a temporary document staging area. This allows SFTP clients to resume a transfer (within a specified timeframe). If the transfer does not resume within the specified amount of time, the Partial Document Clean Up Service removes documents from the staging area and the transfer is no longer available for resumption. A common behavior among SFTP clients before resuming a transfer is to request a list of the directory contents. In response to list requests, the default behavior is for the SFTP Server adapter to return a listing that includes (1) complete documents in the target mailbox and (2) partial documents in the staging area. **Note:** Partial documents are assigned to a particular user. The system only displays partial documents to the user to whom they are assigned. If two documents with the same name exist in both the mailbox and the document staging area, only the partial document in the staging area is displayed in response to a list request. You can change the default behavior by editing the `sftp.properties` file. To enable listing documents that are in the staging area, set listStagedDocuments = True. Default is False. The SFTP Server adapter does not return nonextractable files as part of a directory listing. Once a message becomes nonextractable, it effectively disappears from the SFTP view of the mailbox. The home directory for SFTP is a virtual root mailbox in the application. The mailbox can include both extractable and nonextractable messages. When the SFTP Server adapter accesses the mailbox, only extractable messages are displayed.

System name	SFTP Server Adapter
Permissions	To access the SFTP Server adapter and have full mailbox operations (listing, retrieving, and placing messages), you must have permission to the virtual root (either explicitly assigned or by default). To operate fully on mailboxes in the hierarchy directory, you must have permissions on all mailboxes between the target mailbox and the virtual root and full rights. Rights that can be given on behalf of a user are: write, read, execute, view, and delete. Each right allows specific actions to be performed. By default, a user assigned to a mailbox has all available rights. If a user needs to fully operate on a mailbox at a lower level in the mailbox hierarchy, the user must also have permission and rights on all mailboxes that are between the target mailbox and his virtual root. Rights required for mailbox operations are: • Add a message to a mailbox - Write permission for the Mailbox • Extract message from mailbox -- Read for the Mailbox • List submailbox - Execute for All mailboxes from virtual root to submailbox • List virtual root mailbox - Execute for the Virtual root mailbox • List virtual root mailbox without mailbox execute permission - Execute for the MailboxLoginWithoutVirtual RootPermission • Login if ACL active - Execute for Server Permission • Login to the virtual root mailbox - Execute for Virtual root mailbox • Login to the virtual root mailbox without mailbox execute permission - Execute for MailboxLoginWithout VirtualRootPermission • Move message to mailbox - Write for Destination Mailbox • Remove message from mailbox - Delete Mailbox Restricted operation can be granted to users with a permission 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.
Persistence level	Default

System name	SFTP Server Adapter
Testing considerations	At application startup, attempt to access the SFTP server using a supported SFTP client with the configured IP address and port. Debug information can be found in the SFTP logs. Select Logging Level from the following: • Error - Errors only • Communication Trace - Errors, requests from clients, and responses from the Server adapter, including ACL violations • All - Debugging, all activities

Implementing the SFTP Server Adapter

To implement the SFTP Server adapter, complete the following tasks:

1. Create a configuration of the SFTP Server adapter (or enable the configuration installed with the application and edit parameters as needed).
2. Configure the SFTP Server adapter.

Configuring the SFTP Server Adapter

To configure the SFTP Server adapter:

1. Select **Deployment** > **Services** > **Configuration**.
2. Next to New Service, click **Go**!
3. Select the List View icon, then select the **SFTP Server adapter** from the list. Click **Save**.
4. Click **Next**.
5. Specify field settings:

Field	Description
Name	Name this adapter will have in the application
Description	Description of adapter
Select a Group	None - Do not include this configuration in a group.
Perimeter Server	List of perimeter servers, including local-mode perimeter servers. Required. Default is Node 1 & Local.
Enabled Protocols	Select the protocols to enable for this adapter. Required. Valid values are: • SFTP and SCP • SFTP • SCP Default is SFTP and SCP. **Note:** The SCP option is only available for new configurations of the SFTP Server adapter. If you have a previous version, you can disable it and create a new one to enable SCP or SFTP and SCP.

Field	Description
Host Identity Key	Private/Public key pair used to identify the application SFTP server to remote clients. Required.
SFTP Server Listen Port	The unique port number that the SFTP server should bind to and listen on for connection requests. Cannot be used by any other adapter. Required.
Minimum Number of Threads	A tuning parameter that indicates the minimum number of threads that the perimeter server will use to improve performance. Optional. Default is 3. **Note:** Retain the default value unless instructed otherwise by Sterling Commerce Support.
Maximum Number of Threads	A tuning parameter that indicates the maximum number of threads that the perimeter server will use to improve performance. Optional. Default is 6. **Note:** Retain the default value unless instructed otherwise by Sterling Commerce Support.
Transfer Thread Pool Size	A tuning parameter that indicates the number of permanent transfer threads the server begins with. Once a socket has either been accepted or connected, the socket is registered with a transfer thread. This thread asynchronously performs all the input and output for the socket. If all the permanent threads become fully loaded, additional threads are created to handle additional connections and shut down once they have no sockets to service. Optional. Default is 2.
Channels per Transfer Thread	A tuning parameter that indicates the number of channels available for each transfer thread. Set maximum number of SelectableChannels that can be assigned to the accept, transfer, and connect selectors. Value of 1 effectively makes server behave in thread-per-connection mode. Optional. Default is 400.
Maximum Authentications	The maximum number of failed authentication attempts a user is allowed before the session is ended. Optional. Default is 3.
Session Timeout (seconds)	The number of seconds each session is allowed to last. Required. Valid value is any number between 1 and 9,999,999. Default is 120,000. **Note:** If the timeout is reached during a transfer, the session will be closed immediately after the transfer completes.
Resumption Timeout (hours)	Timeout value for the incomplete document before it is purged. Required. Valid value is any number between 1 and 9,999,999. Default is 48.

Field	Description
Compression	Specifies whether data is to be compressed, which reduces the amount of data transmitted as the file is copied from one node to another. The file will be automatically decompressed at the destination. Optional. Valid values: None, ZLIB.
PreferredCipher	The cipher the server prefers to use for both client to server and server to client stream encryption. Optional. Default is blowfish-cbc. Valid values are: • 3des-cbc • blowfish-cbc • aes256-cbc • aes192-cbc • aes128-cbc • cast128-cbc • twofish256-cbc • twofish192-cbc • twofish128-cbc
PreferredMAC	The MAC the server prefers to use for stream encryption. Optional. Valid values are: • hmac-sha1 • hmac-md5 Default is hmac-sha1
Required Authentication	Specifies the type of authentication required for the adapter. Required. Valid values are: • Password or Public Key (default) • Password • Public Key • Password and Public Key
Maximum Logins	Maximum number of logins the adapter may have active at any point of time. Use this to limit the total number of users allowed to access a server at any one time. This can be used to manage server performance. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999.
Maximum Logins Per User	Maximum number of logins each user may have active on this adapter at any point of time. Use this to limit users who want to make many connections at the same time to ensure bandwidth is shared among users. If no value is specified, logins are unlimited. Optional. Valid value is any integer to 9999999999.

Field	Description
Document Storage Type	Select whether documents will be stored on the file system, the database, or the system default. Required. Valid values are: • File System (default) - Default value when the application is installed, but it can be changed. Contact your system administrator to see if the default has been changed. • Database - Body of the request document will be stored in the database. • System Default - If your system administrator has changed the installed default of File System, this ensures that the correct location is used.
Support for concurrent duplicate-named file transfers	Allows sending the files with same name, concurrently to the same Mailbox using the same username. It also allows partners to receive multiple duplicate files with the same name, concurrently, but with different content. Valid values are: • Limited (resume of file transfers supported) - Default. This is the default value. The file transfer can be resumed if transfer fails from the point of failure. You cannot transfer duplicate files with the same name concurrently to the same Mailbox and the same username. • Full, concatenate duplicate-named files on a GET (resume of file transfers not supported) - Supports sending duplicate files with the same name concurrently to the same Mailbox using the same username. The files with the same name are concatenated on a GET operation. You cannot resume broken file transfers. • Full (resume of file transfers not supported) - Supports sending duplicate files with the same name concurrently to the same Mailbox using the same username. The files with the same name are not concatenated on GET operation. The GET operation of a duplicate file would retrieve the latest extractable message from the Mailbox. You cannot resume broken file transfers.
Should the adapter be restricted to a certain group of users?	Select Yes or No to indicate whether to restrict specific users and groups to access the SFTP server. Required. Default is No. If Yes, select Users and or Groups from the lists on subsequent pages.

Field	Description
Should users start in the directory that matches their user name upon login?	Places the user, upon logging in, into a directory (mailbox in the application) that corresponds to his or her user ID. Valid values are: • 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. **Note:** Do not select Yes if there is any chance that users of your application might have user IDs that differ only by case (example: jsmith and JSmith). Unlike user IDs, mailbox names in this application are not case-sensitive. • No - The user is placed in the virtual root directory.
Users	Select a list of users who are granted permission to access the server.
Groups	Select a list of groups who are granted permission to access the server.
Extractable Count	The number of times the message can be extracted. Cannot be specified in conjunction with Extractable or Extractable For. Optional. Valid value is any integer.
Extractable For	A counter indicating the length of time (in days, hours and minutes) the message can be extracted.Cannot be specified in conjunction with Extractable or Extractable Count. Optional. Format is dddhhmm.
Extractable	A yes or no value indicating if this message can be extracted. Cannot be specified in conjunction with Extractable Count or Extractable For. Optional.

6. On the Confirm screen, ensure that **Enable service for Business Process** is selected. Click **Finish**.

Correlations and Document Tracking

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

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

Key	Values
Username	username

Activity Monitoring for the SFTP Server Adapter

The SFTP Server adapter creates activity monitoring records for the following activities:

- Active sessions (connections to clients)
- In progress PUTs display the data transferred in kbps with a progress indicator
- In progress GETs display the data transferred in kbps

To view the records, select **Business Processes** > **Current Activities** > **SFTP Server Adapter**.

Chapter 4. Build 5004 or Higher

Import Service

The following table provides an overview of the Import service:

System name	Import Service
Graphical Process Modeler (GPM) category	All Services
Description	This service is used in a business process to automatically import application resources exported using the Resource Manager, including: • SAP application configurations • Translation maps • Trading partner data (packages, identities, contracts, envelopers, and code lists) • Business processes • Service configurations • XML schemas • XSLT stylesheets • Web templates • Web resources (JSP files, JavaScript files, HTML files, XML files, image files, property files, stylesheets, and custom defined files)
Business usage	In a hub and spoke relationship, a hub company could use this service to programatically update information on their trading partners' systems.
Usage example	A hub needs to update its trading partner information with all of its spokes. At the same time, it plans on rolling out new XML schemas and translation maps. The hub creates an installable bundle interactively by means of the resource manager. The bundle is sent to the trading partners affected. The bundle is picked up by the trading partners and processed by a business process set up as the updates from trading partner hub x. This process includes the Import service. The service checks the security context and, assuming it is correct, opens the bundle and updates the local system with the updates automatically.
Preconfigured?	Yes
Requires third party files?	No
Platform availability	All supported platforms

System name	Import Service
Related services	This service is designed to work in conjunction with a transport type service. The transport service brings the resources into the local system.
Application requirements	Before using this service, a security context for the installable resource bundle must have been created using the Security Context utility.
Initiates business processes?	No
Invocation	Event driven
Business process context considerations	The Import service configuration may contain context and identity values for a security context, which are used during the business process to fetch the passphrase for verification if the file being imported contains encrypted data. If a passphrase is required but the values are not for the correct passphrase or no security context information is available (either the values from the service configuration or the passphrase value stored for each configuration in the application database), the Import service will fail.
Returned status values	• Success - The service completed successfully. • 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. • All resources defined in the installable bundle will be installed. • Any existing resources will be updated, and the version number incremented. • The installed resource will become the default, if applicable.
Persistence level	Full
Testing considerations	Export a set of resources from the application to a file called Export.xml. Import these resources in to another application server. Check the status report. There must not be any errors and it should be possible to test the imported resources.

How the Import Service Works

The Import service exercises the same functionality as the Import Resources option in the Resource Manager, with one exception-the service has no user interaction, so the service does not ask for confirmation of options. When using the Import Service, all available resources are imported, and all imported versions are set as the defaults, where applicable.

You can create a security context for an installable bundle, which can prevent unauthorized users from creating or updating resources.

The Import Service works with the Security Context utility in the application. The utility is called securityContext.sh (for Unix) or securityContext.cmd (for Windows). It is located in the bin directory of your application installation. This is an example of how the security context is used:

1. A developer at company A exports a resource bundle to be sent to company B, where the bundle will be imported. If required for the type of resource to be exported, the developer creates a passphrase for the resource bundle as a part of the export process.
2. After the export is complete, the developer sends the resource bundle to the company B system administrator, and also informs the system administrator of the passphrase.
3. The system administrator at company B uses the Security Context utility to enter the passphrase into the application database and to create a security context.
4. The system administrator passes the resource bundle and the name of the security context to a developer.
5. The developer configures the Import service in the GPM, using the context and identity values from the security context that the system administrator provided.

For more information about creating a security context, see *Using the Security Context Utility*.

Implementing the Import Service

To implement the Import Service, complete the following tasks:

1. After receiving a resource bundle from a trading partner, create a security context for it. For information, see *Using the Security Context Utility*.
2. Create an Import Service configuration. For more information, see *Managing Services and Adapters*.
3. Configure the Import Service.
4. Use the Import Service in a business process.

Configuring the Import Service

To configure the Import Service, you must you must specify settings for the following fields in your application:

Field	Description
Backup	Identifies the path where the backup is saved. If the path is invalid during backup, the file is written to <install>/tmp and a message is added to the Import Report indicating the location. If the parameter is not specified, then the backup is not generated. **Note:** A back up file may not be created if the file name of the document contains an underscore (_) character, for example, file_name. If the file name has an underscore character, it is recommended to manually back up the file before importing.
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 application installation.

Action	Description	Usage

list	Lists all security contexts available.	Unix:
		`installdir/bin>securityContext.sh list_context`
		Windows:
		`installdir\bin>securityContext.cmd list_context`

set	Updates the database in application with the new context. Takes three parameters:
	• context
	• identity
	• passphrase
		Unix:
		`installdir/bin>securityContext.sh set context identity passphrase`
		Windows:
		`installdir\bin>securityContext.cmd set context identity passphrase`
		Returns the following message:
		Context saved.

get	Returns the passphrase value for the context.
	Takes two parameters: context and identity.
		Unix:
		`installdir//bin>securityContext.sh get context identity`
		Windows:
		`installdir\bin>securityContext.cmd get context identity`
		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 application system, and uses the list_context action to find out. There are three contexts on the system: MaxxMart, Taylor, and Zapf.

```
install_dir\bin>securityContext.sh list_context
Contexts:
MaxxMart,Taylor,Zapf
```

In the third example, Jill wants to know what the passphrase is for the security context named Taylor that has an identity of jan20. She uses the get action and finds that the passphrase is thunder.

```
install_dir\bin>securityContext.sh get Taylor jan20
Taylor,jan20,thunder
```

Business Process Example 1

The following example illustrates using the Import service in a business process to import a resource file called dec19 from Company_x:

```
<process name="ImportService">
  <sequence>
    <operation>
      <participant name="ImportService"/>
      <output message="Xout">
<assign to="Context">company_x</assign>
<assign to="Identity">dec19</assign>
         <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
         <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Business Process Example 2

The following example illustrates using the Import service in a business process to import a resource file called april1 from RomansFloorsAndMore, using the Keep Existing Control Numbers option (specifying that for existing envelopes and control numbers, control number values in the import file will not be imported if a version of an envelope or control number being imported already exists in the system). The import process will overwrite the value specified in the import file with the existing control number value for that envelope or control number:

```
<process name="ImportServiceWithKeepExistingControlNumbers">
<sequence>
    <operation>
      <participant name="ImportService"/>
      <output message="Xout">
 <assign to="Context">RomansFloorsAndMore</assign>
 <assign to="Identity">april1</assign>
 <assign to="KeepExistingControlNumbers">True</assign>
 <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
         <assign to="." from="*"></assign>
      </input>
    </operation>
  </sequence>
</process>
```

Viewing the Import Service Status Report

Once you have imported resources using the Import service in a business process, it is a good idea to check that all resources were imported successfully. Also, you may need to use the report for troubleshooting if the service and business process fail. You can view the status report from the Business Process Detail page.

To view the report Business Process Detail page, complete the following steps:

1. From the Business Process menu, select **Monitor** > **Current Processes**. Current business processes are displayed in a list.
2. Click the Instance ID next to the desired business process. The Business Process Detail page for that business process displays.
3. In the Status Report column, click the Info icon for the Import service. The report is opened in another window. The status of each resource that the service attempted to import is shown, which enables you to verify whether each was successfully imported or not.

Some reasons that the Import service might fail (which will cause the business process to fail) in situations where passphrase (Context/Identity) is required are:

- Invalid passphrase (the passphrase in the database doesn't match the passphrase in the resource bundle).
- No passphrase in situation where passphrase is required (possibly there was no security context created for this resource bundle).
- Either the context or identity value in the Import service configuration used in the business process is wrong or was left blank.

The following is a sample status report for an import bundle. There is one error for a resource that could not be imported (transport account password):

```
Name: UpdateTPInfo     Instance ID:1053     Service Name: Import Service
  Status report on 2004-03-12 14:45:19.16 for service: Import
Packaging :: packaging_1079119091618 :: update :: SUCCESS :: Resource successfully
imported.
Identity :: MaxxMart :: update :: SUCCESS :: Resource successfully imported.
Transport :: HTTP Transport :: create :: Message :: Error decrypting transport
account password...value will be stored as it was in import file.
Transport :: HTTP Transport :: update :: SUCCESS :: Resource successfully imported.
Document Exchange :: MaxxMart Doc Exchange :: update :: SUCCESS :: Resource
successfully imported.
Delivery Channel :: ABCD :: update :: SUCCESS :: Resource successfully imported.
Profile :: MaxxMart1 :: update :: SUCCESS :: Resource successfully imported.
End of report
ImportService stayed in queue 7  ms
```

PGP Package Service

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Package service, in conjunction with the PGP Server Manager, enables you to encrypt and digitally sign documents using PGP.

The following table provides an overview of the PGP Package service:

System name	PGP Package service
Graphical Process Modeler (GPM) category	All Services
Description	This service encrypts and digitally signs a document based on the Open PGP standard, using public key or conventional cryptography.
Business usage	Use this service to encrypt and sign a document in the document area of process data.
Usage example	A business process is executed to encrypt and sign a document, based on the information stored in a PGP profile.

System name	PGP Package service
Preconfigured?	Yes. A configuration called PGP Package Service is installed with Application.
Requires third-party files?	No
Platform availability	All supported Application platforms, with the following restrictions: **For NAI McAfee eBusiness Server 8.1** • IBM AIX 4.2 or later • HP-UX 10.20 or later • Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) • SuSE Linux for IBM S/390 and IBM Zseries **For NAI McAfee eBusiness Server 8.5** • Solaris 9 or later **For NAI McAfee eBusiness Server 8.5.1** • Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) • Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) • Microsoft Windows Server 2003 • Microsoft Windows XP Professional Version 2002 Service Pack 2 **For Massachusetts Institute of Technology (MIT) Command Line Freeware** • Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 • UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later **For PGP Corporation PGP® Command Line 9.5** • Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 (SP 1) Microsoft Windows 2000 (SP 4) • UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 • Mac OS X 10.4 or greater

System name	PGP Package service
Related adapters and services	The PGP Package service works with the following services: • Command Line Adapter 2 • PGP Unpackage service
Application requirements	Before using this service, install one of the following: • McAfee E-Business Server (version 8.1, 8.5, 8.5.1, or 8.6) from Network Associates Technology, Inc. • PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) • PGP Command Line (version 9.5) from PGP Corporation **Note:** Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	Basic statuses are: • 0 - Success • 1- Error See *Advanced Status Messages* for a list of advanced statuses. Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrorCode].
Restrictions	None
Persistence level	None

System name	PGP Package service

Testing considerations	Create the profile in the PGP Server Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.
	
	A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.
	
	To start the remote adapter implementation of the command line adapter:
	1. Locate the client jar (CLA2Client.jar in Install_DIR>/<client>/<cmdline2>) that contains all the necessary classes.
	2. Move the client jar to the machine that has the PGP server installed.
	3. Start the remote adapter implementation using the following command:
	`java -jar CLA2Client.jar <port> [debug]`
	
	For example:
	`java -jar CLA2Client.jar 15699 debug`
	
	Note: The [debug] option is not required.

Implementing the PGP Package Service

To implement the PGP Package service, complete the following tasks:

1. Activate your license for the PGP Package service. See *Managing Services and Adapters*.
2. Create a PGP profile, using the Application PGP Server Manager. See *PGP Server Manager*.
3. Create a PGP Package service configuration. See *Managing Services and Adapters*.
4. Configure the service. See *Configuring the PGP Package Service*.
5. Use the PGP Package service in a business process.

Configuring the PGP Package Service

Before configuring, consider the following:

- public_user (if using Public Key Cryptography) or conv_keymap_name (if using Conventional Cryptography) must be present for PGP Package service to perform encryption.
- secret_keymap_name must be present for PGP Package service to perform signing.
- To perform encryption and signing, a combination of both the previous statements applies.

- If public_user and conv_keymap_name appear in the same business process, public key encryption will take precedence.

To configure the PGP Package service, specify settings specify the settings for the fields in the GPM. These fields are described in the following table:

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files used for encryption and signing will be read from or written to. Optional if the cmdline2svcname field is defined in the Command Line Adapter 2.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the cmdline2svcname field is defined in the Command Line Adapter 2.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the cmdline2svcname field is defined in the Command Line Adapter 2.
profile_name	Name of PGP profile from the PGP Server Manager. Required.
compress	Compression to be done before encryption or signing. Valid value is On. Default is On. Required for encryption and signing.
public_user	User name or key ID in the public key ring. Required for encryption (public key cryptography).
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for signing (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for encryption (conventional cryptography).
conv_cipher	The symmetric cipher to use when performing a conventional encryption operation (that is, conv_keymap_name is used). Valid values are: IDEA, CAST5, 3DES, AES128, AES196, AES256, Twofish. Default is IDEA. Optional.
DocumentId	The document identifier referenced to the document to be processed specifically. The default document for processing is the primary document. Optional.
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
ascii_armor	Whether to encode the file with McAfee E-Business Server's base-64 encoding (ASCII-armored format). Valid values are On and Off. Default is On. Optional.

Field	Description
textmode	Whether the input data is ASCII text and should be converted to canonical new lines before encryption. Valid values are On and Off. Default is Off. Optional.
outputfilename	Output file name. For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc. For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of the document and is appended with the following: • *.asc during normal encryption • .exe during sda process • .pga during pgparchive process Optional.
pgp_partner_name	The partner name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected partner profile. Required if you specify a value in the pgp_sponsor_name parameter. The values you specify in the GPM override the values you specify in the profile.
pgp_sponsor_name	The sponsor name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected sponsor profile. Required if you specify a value in the pgp_partner_name parameter. The values you specify in the GPM override the values you specify in the profile.
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.

Field	Description
clearsig	Generates a signed message that can be read without PGP. The recipient must still use PGP to verify the signature. Unencrypted PGP-signed messages have a signature certificate pre-pended in binary form. The signed message is compressed. Therefore, it is unreadable by humans even though it is not encrypted. Cannot be used with EncryptAndSign on the command line. If you enable clearsig, it is recommended you enable ascii_armor and textmode also. Valid values are On and Off. Default is Off. Optional.
info	How much information is returned. Valid values are: • Quiet - Only displays error messages. Not applicable to PGP Command Line. If selected defaults to normal mode. • Normal - Displays warnings and error messages. Default. • Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). If selected with other versions, defaults to normal mode. • Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line. If selected, defaults to normal mode. Optional.

Field	Description
sda	Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified. Creates a self-decrypting executable file, which is conventionally encrypted using a passphrase. The resulting file can be decrypted by double-clicking it and entering the passphrase. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed. SDA files can be created with any platform that McAfee E-Business Server (version 8.1 or later) supports, but can be executed only on Windows platforms. To create sda files with PGP Command Line (version 9.5), set the target_platform parameter (described later in this table). The default file extension is .exe. **Note:** The sda file cannot exceed 4 GB after compression. Valid values are On and Off. Default is Off. Optional.
pgparchive	Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified. Creates a file that can be decrypted using the archive reader, which can be redistributed freely. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed. The default extension is .pga. Valid values are On and Off. Default is Off. Optional.
discard_paths	Applicable only with sda or pgparchive. Strips relative path information form the list of files in a sda or pgparchive. During the decryption of the archive, the files are placed in the current directory instead of in subdirectories of the current directory. Optional.

Field	Description
target_platform	Applicable only with PGP Command Line (version 9.5) and sda. Specifies the platform an sda file can be decrypted on. Valid values are: • win32 • linux • solaris • aix • hpux • osx Default is the current platform. Optional.

Parameters Passed from Service to BP

The following table contains the parameters that are passed from the PGP Package service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are: • ENCRYPT • ENCRYPT_SIGN • SIGN Required.
FileName (PGP/FileName)	Name of the file being processed. Required.
inputFileNamePkg (PGP/inputFileNamePkg)	Name of the file contained in the PGP package. Optional.
Document (PGP/Document)	The processed document is placed in Process Data - not as Primary Document. The attribute is the SCIObjectID, which enables a hyperlink for viewing the content of the processed document. Required.
DocumentId (PGP/DocumentId)	Document identifier of the document. Required.
Status (PGP/Status)	Process status. Valid values are Success and Error. Required.
ErrorCode (PGP/ErrorCode)	Value returned from executing PGP commands. Displayed when the Status is Error. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. Displayed when the Status is Error. Optional.

Business Process Example - Encrypt Operation (Public Key Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area. The profile is based on PGP107. In this example, you use the default Command Line2 adapter configuration, PGPCmdlineService, to execute the encrypt command. You want to use the working directory, remote name and port stated in the BPML. Therefore, these values override the pre-configured values in PGPCmdLineService. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption.

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
              <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="public_user">0x2343</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt Operation (Conventional Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area of process data. The profile is based on PGP107. In this example, you use the Command Line2 adapter configuration, MyCLA2, to execute the commands. The remote name, port, and working directory are pre-configured in the service configuration. The value of conv_keymap_name, Conv_abc_tp, which must be in the profile's conventional key map, is used for conventional encryption:

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="conv_keymap_name">Conv_abc_tp</assign>
              <assign to="conv_cipher">CAST5</assign>
              <assign to="cmdline2svcname">MyCLA2</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt and Sign Operation (Public Key Encryption)

The following business process uses the PGP Package service to encrypt and sign the primary document in the document area. For signing, you need to pass in the secret_keymap_name, which must be in the PGP107 profile's secret key map. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption. In this example, you choose not to compress the document before signing and encryption.

```
<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">off</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="public_user">0x2343</assign>
              <assign to="secret_keymap_name">my_secret</assign>
</output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt and Sign Operation (Conventional Encryption)

The following business process uses PGP Package Service to encrypt and sign the Primary Document in the document area. For signing, the user needs to pass in the secret_keymap_name, which must be present in the PGP107 profile's Secret Key Map. The value of conv_keymap_name, Conv_abc_tp, which must be present in the Profile's Conventional Key Map, is used for conventional encryption. The user chooses not to compress the document before signing and encryption.

```
<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">off</assign>
              <assign to="workingDir">/localsvr/share/tmp</assign>
              <assign to="remoteName">nn.nnn.nn.nnn</assign>
              <assign to="remotePort">xxxxx</assign>
              <assign to="conv_keymap_name">Conv_abc_tp</assign>
              <assign to="conv_cipher">CAST5</assign>
              <assign to="secret_keymap_name">si_secret</assign>
              <assign to="." from="*"></assign>
     </output>
     <input message="Xin">
        <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt Operation (Public Key Encryption) Using a Specific Document ID

The following business process uses the PGP Package service to encrypt a document, with the document ID columbia:1774b9b:feaea8ae12:-6ea8 in the document area.

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One"> PGPPackageService
      <participant name="PGPPackageService"/>
      <output message="Xout">
              <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="public_user">0x2343</assign>
              <assign to="DocumentId">columbia:1774b9b:feaea8ae12:-6ea8</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Sign Operation

The following business process uses the PGP Package service to sign the primary document in the document area.

```
<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="secret_keymap_name">my_secret</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - OnFault Handling

The following business process shows the onFault handling for the PGP Package service.

```
<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
   <assign to="profile_name">PGP107</assign>
   <assign to="compress">on</assign>
   <assign to="workingDir">/localsvr/share/tmp</assign>
```

```
   <assign to="remoteName">nn.nnn.nn.nnn</assign>
   <assign to="remotePort">12345</assign>
   <assign to="secret_keymap_name">si_secret</assign>
   <assign to="." from="*"></assign>
     </output>
     <input message="Xin">
        <assign to="." from="*"></assign>
     </input>
    </operation>
    <assign to="Status">The file is signed successfully</assign>
    <onFault>
      <assign to="Status">General Error Occurred</assign>
    </onFault>
    <onFault code="[PGPErrorCode] Signature Check error">
      <assign to="Status">Incorrect signature</assign>
    </onFault>
  </sequence>
</process>
```

Business Process Example - PGP Partner and PGP Sponsor

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

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

Advanced Status Messages

The following table contains exit codes from the McAfee E-Business Server and PGP Command Line Freeware. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error

Status	Description
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.
160	Complete failure during a file wipe.
161	Partial fail, partial success during a file wipe (one file wiped, one not, for example).
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

The following table contains errors that result from the PGP Package service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId	The DocumentId value given in the BPML is incorrect.
Fail to get data from Primary Document.: There is no Primary Document	Primary Document is mandatory.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Server Manager	The profile_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap	The secret_keymap_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap	The conv_keymap_name value given in the BPML is incorrect.

PGP Unpackage Service

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Unpackage service, in conjunction with the PGP Server Manager, enables you to decrypt documents and verify their signatures.

The following table provides an overview of the PGP Unpackage service:

System name	PGP Unpackage service
Graphical Process Modeler (GPM) category	All Services
Description	This service is used to decrypt and verify the signature of a document based on the Open PGP standard, using a public key or conventional cryptography.
Business usage	Use this service to decrypt or verify the signature of the document in the document area.
Usage example	A business process is executed to decrypt or verify the signature of the document based on the PGP profile. See *PGP Server Manager*.
Preconfigured?	Yes
Requires third-party files?	No

System name	PGP Unpackage service
Platform availability	All supported Application platforms, with the following restrictions: **For NAI McAfee eBusiness Server 8.1** • IBM AIX 4.2 or later • HP-UX 10.20 or later • Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) • SuSE Linux for IBM S/390 and IBM Zseries **For NAI McAfee eBusiness Server 8.5** • Solaris 9 or later **For NAI McAfee eBusiness Server 8.5.1** • Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) • Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) • Microsoft Windows Server 2003 • Microsoft Windows XP Professional Version 2002 Service Pack 2 **For Massachusetts Institute of Technology (MIT) Command Line Freeware** • Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 • UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later **For PGP Corporation PGP® Command Line 9.5** • Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 (SP 1) Microsoft Windows 2000 (SP 4) • UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 • Mac OS X 10.4 or greater

System name	PGP Unpackage service
Related adapters	The PGP Unpackage service works with the following services: • Command Line Adapter 2 • PGP Package service
Application requirements	Before using this service, install one of the following: • McAfee E-Business Server (version 8.1, 8.5, 8.5.1, or 8.6) from Network Associates Technology, Inc. • PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) • PGP Command Line (version 9.5) from PGP Corporation **Note:** Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	Basic statuses are: • 0 - Success • 1- Error See *Advanced Status Messages* for a list of advanced statuses. Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrorCode].
Restrictions	None
Persistence level	None

System name	PGP Unpackage service

Testing considerations	Create the profile in the PGP Server Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.
	
	A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.
	
	To start the remote adapter implementation of the command line adapter:
	1. Locate the client jar (CLA2Client.jar) that contains all the necessary classes in the following directory:
	`install_DIR>/<client>/<cmdline2>`
	2. Move the client jar to the machine that has the PGP server installed.
	3. Start the remote adapter implementation using the following command:
	`java -jar CLA2Client.jar <port> [debug]`
	For example:
	`java -jar CLA2Client.jar 15699 debug`
	Note: The [debug] option is not required.

Implement the PGP Unpackage Service

To implement the PGP Unpackage service, complete the following tasks:

1. Activate your license for the PGP Unpackage service.
2. Create a PGP profile, using the PGP Server Manager.
3. Create a PGP Unpackage service configuration.
4. Configure the PGP Unpackage service.
5. Use the PGP Unpackage service in a business process.

Configure the PGP Unpackage Service

Before configuring the PGP Unpackage service, consider the following:

- If the secret_keymap_name and conv_keymap_name parameters are not present, the PGP Unpackage service will verify the signature of the document only.
- If one of the keymap_name parameters is present, it will use the information of the keymap_name to decrypt.
- If there is a signature in the document, the verification of the signature will be done automatically.

To configure the PGP Unpackage service, specify the settings for the fields in the GPM. These fields are described in the subsequent table.

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files for decryption or verification will be read from or written to. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the cmdline2svcname field is defined in the Command Line 2 adapter. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the cmdline2svcname field is defined in the Command Line 2 adapter. You must set this parameter in this field or in the associated Command Line 2 adapter configuration.
profile_name	The name of PGP profile. Required.
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for decryption (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for decryption (conventional cryptography).
DocumentId	The document identifier for the document to be processed. The default document for processing is the primary document. Optional.
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
outputfilename	Output file name. For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc. For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of a document and is appended with the following: • *.asc during normal encryption • .exe during SDA process • .pga during pgparchive process Optional.

Field	Description
pgp_partner_name	The partner name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected partner profile. Required if you specify a value in the pgp_sponsor_name parameter. The values you specify in the GPM override the values you specify in the profile.
pgp_sponsor_name	The sponsor name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected sponsor profile. Required if you specify a value in the pgp_partner_name parameter. The values you specify in the GPM override the values you specify in the profile.
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.
info	How much information is returned. Valid values are: • Quiet - Only displays error messages. Not applicable to PGP Command Line (version 9.5). If selected, defaults to normal mode. • Normal - Displays warnings and error messages. Default. • Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command LIne (version 9.5). If selected with other versions, defaults to normal mode. • Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line (version 9.5). If selected, defaults to normal mode. Optional.

The following table contains the parameters that are passed from the PGP Unpackage service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are DECRYPT and VERIFY. Required.
FileName (PGP/FileName)	The name of the file which is being processed. Required.
Document PGP/Document()	The processed document is placed in Process Data - not as Primary Document. The attribute is the SCIObjectID, which allows the user to click on it for viewing the content of the processed document. Required.
DocumentId (PGP/DocumentId)	The document identifier of the document. Required.
Status (PGP/Status)	The status shows if this process has completed successfully or failed. Valid values are Success and Error. Required.
ErrorCode PGP/ErrorCode()	This is the exit value returned from executing PGP commands. This will be shown when the Status is `Error'. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. This will be shown when the Status is `Error'. Optional.

Business Process Example - Decrypt Operation (Public Key Decryption)

The following business process uses the PGP Unpackage service to decrypt the primary document in the document area. The profile is based on PGP107. In this case, the default Command Line 2 adapter configuration, PGPCmdlineService, is used to execute the decrypt command. It uses the working directory, remote name and port stated in the business process. Therefore, these values will override any pre-configured values in PGPCmdlineService.

```
<process name="PGP_Decrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
               <assign to="." from="*"></assign>
               <assign to="profile_name">PGP107</assign>
               <assign to=" secret_keymap_name"> my_secret </assign>
               <assign to="workingDir">/server1/tmp</assign>
               <assign to="remoteName">00.000.00.000</assign>
               <assign to="remotePort">12345</assign>
     </output>
     <input message="Xin">
               <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Verify Operation

The following business process uses the PGP Unpackage service to verify the primary document in the document area. The profile is based on PGP107. In this case, the Command Line 2 adapter configuration called MyCLA2 is used to execute the commands. The remote name, port and working directory have been pre-configured in the service configuration. Therefore, they are not required in the business process.

```
<process name="PGP_Verify">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
               <assign to="." from="*"></assign>
               <assign to="profile_name">PGP107</assign>
               <assign to="cmdline2svcname">MyCLA2</assign>
     </output>
     <input message="Xin">
               <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - OnFault Handling

The following business process shows onFault handling with the PGP Unpackage service.

```
<process name="PGP_Decrypt">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPUnPackageService "/>
      <output message="Xout">
               <assign to="." from="*"></assign>
               <assign to="profile_name">PGP107</assign>
               <assign to=" secret_keymap_name"> si_secret </assign>
               <assign to="workingDir">/server1/tmp</assign>
               <assign to="remoteName">00.000.00.000</assign>
               <assign to="remotePort">12345</assign>
     </output>
     <input message="Xin">
               <assign to="." from="*"></assign>
     </input>
    </operation>
               <assign to="Status">The file is decrypted successfully</assign>
    <onFault>
               <assign to="Status">General Error Occurred</assign>
    </onFault>
    <onFault code="[PGPErrorCode] Decryption error">
               <assign to="Status">Decryption error</assign>
    </onFault>
  </sequence>
</process>
```

Business Process Example - PGP Partner and PGP Sponsor

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

```
<process name="use_partner_sponsor">
  <operation name="PGP Unpackage Service">
    <participant name="PGPUnpackageService"/>
       <output message="PGPUnpackageServiceTypeInputMessage">
```

```
      <assign to="pgp_partner_name">partner</assign>
      <assign to="pgp_sponsor_name">sponsor</assign>
      <assign to="profile_name">pgp</assign>
      <assign to="." from="*"></assign>
    </output>
    <input message="inmsg">
      <assign to="." from="*"></assign>
    </input>
  </operation>
</process>
```

Advanced Status Messages

Exit Codes from E-Business Server and PGP Command Line Freeware

The following table contains exit codes from E-Business Server and PGP Command Line Freeware. The content of the description field will be displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

Exit Codes from PGP Command Line - PGP Corporation

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the description field will be displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.

Status	Description
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.
160	Complete failure during a file wipe.
161	Complete failure during a file wipe.
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

Errors During Validation

The following table contains errors that result from the PGP Unpackage service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId	The DocumentId value given in the bpml is incorrect.
Fail to get data from Primary Document	There is no Primary Document. Primary Document is mandatory.
You must enter one of these BPML Params: 'public_user' or 'secret_keymap_name' or 'conv_keymap_name'	Either one of the BPML Parameters must be present for PGP to encrypt, sign or encrypt and sign.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Server Manager	The profile_name value given in the bpml is incorrect.

Status	Description
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap	The secret_keymap_name value given in the bpml is incorrect.
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap	The conv_keymap_name value given in the bpml is incorrect.

XSLT Service

The following table provides an overview of the XSLT service:

System name	XSLT Service
Graphical Process Modeler (GPM) categories	All Services, Translation, Sync Mode, Transactional Mode
Description	Enables you to use XSLT style sheets in Application. The XSLT service performs transformation of an XML document from specified location (primary document or process data) using selected XSLT. It can also do input XML validation.
Business usage	Could be used to perform any sort of transformation on XML documents where the output is specified in the XSLT itself. The service could be used to produce static HTML page using data from input XML.
Usage example	There is an automotive parts ordering service, where the manufacturer receives an XML document (containing an order) from a supplier. The manufacturer can set up a business process that uses this service to transform the XML to another format that their system can understand.
Preconfigured?	Yes
Requires third party files?	You may need to check in XSLT stylesheets.
Platform availability	All supported Application platforms
Related services	No
Application requirements	No
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	The service uses and modifies the business process context document content.
Returned status values	• Basic status Success - Transformation was successful. • Basic status Error - Errors were encountered during transformation or transformation could not be performed. See the report contained in the business process context status report for further detail.

System name	XSLT Service
Restrictions	None
Persistence level	None
Testing considerations	Problems to look for would be malformed or invalid XML and XSLT. Verify valid output of the transformation. If the transformer fails to allocate a field in XML data during transformation, it does not report it as an error; instead, leaves it blank.

Requirements

Before you configure the XSLT service in the GPM, you must:

- Be proficient in XSLT.
- Check in any XSLT style sheets you want to use. See *Checking In an XSLT Style Sheet*.

Memory Requirements

To process large files, the physical memory allocated to Application must be at least six times the size of the file to be processed. See *Performance Tuning Utility* to allocate more physical memory to Application.

The physical memory allocated to Application divided by a memory factor indicates the maximum file size that can be processed by the service. However, it depends on the load of the machine at the time of processing.

Let us consider the following example to understand the concept of memory to be allocated to Application and the memory factor to process a file.

Memory Allocated to Application	Memory Factor	Approximate File Size that can be Processed (Memory Allocated to Application / Memory Factor)
<= 1024 MB	8	Assuming that memory allocated to Application is 1024 MB, the XSLT service can process a 128 MB (1024 / 8) file.
> 1024 MB and <= 1536 MB	7	Assuming that memory allocated to Application is 1280 MB, the XSLT service can process a 182 MB (1280 / 7) file.
> 1536 MB	6	Assuming that memory allocated to Application is 1536 MB, the XSLT service can process a 256 MB (1536 / 6) file.

Implementing the XSLT Service

To implement the XSLT service, complete the following tasks:

1. Activate your license for the XSLT service.
2. Create an XSLT service configuration, if necessary. Application provides a standard configuration of the XSLT service for you (named XSLTService). You do not need to create one. However, you may choose to create a unique XSLT service configuration.

3. Configure the XSLT service.
4. Check in any XSLT style sheets. See *Managing XSLT Style Sheets*.
5. Use the XSLT service in a business process.

Configuring the XSLT Service

Application provides a standard configuration of the XSLT service for you (named XSLTService). You do not need to create one. However, you may choose to create a unique XSLT service configuration.

To configure the XSLT service, you must configure the following fields in the GPM:

Field	Description
Config	Name of the service configuration.
additional_xslt_parms	Where you specify additional parameters in the XSLT style sheet. Click this field, then the ellipses to enter key-value pairs in the Value of additional_xslt_parms dialog box. Click the icon button to the right to use the XPath Expression Builder.
input_pd_xpath	Location of the input XML in the process data document using XPath, if the XML document comes from process data. Required if the value for xml_input_from is process data. Click this field, then the ellipses to enter key-value pairs in the Value of input_pd_xpath dialog box. Click the icon button to the right to use the XPath Expression Builder.
xml_input_from	Where the service should receive the XML document from, either primary document or process data.
xml_input_validation	Select No validation if you do not want to validate the input XML document or select dtd or schema to use either one to validate the input XML document.
xslt_name	Previously checked in XSLT style sheet that you want to use.
load_from_classpath	If set to true, the system will look for the template (named by the xslt_name parameter) on the classpath. Valid values are true and false. Default is false. Optional.
incremental_transform	A performance feature of the transformer. If set to true and physical memory allocated to Application increased, XSLT service can process large files according to the memory requirements. Valid values are True and False. Default is False. Optional.

Managing XSLT Style Sheets

Managing XSLT style sheets involves the following tasks:

- Checking In an XSLT Style Sheet
- Checking In an XSLT Style Sheet Using the Text Editor
- Searching for an XSLT Style Sheet
- About Search Results
- Editing an XSLT Style Sheet
- Checking In an Updated Version of an XSLT Style Sheet
- Checking Out an XSLT Style Sheet
- Enabling or Disabling an XSLT Style Sheet
- Specifying a Default XSLT Style Sheet

Checking In an XSLT Style Sheet

To use XSLT style sheets in Application, you must first check them in.

To check in an XSLT style sheet to Application:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Check-in**, click **Go**!
3. Type the name of the XSLT style sheet.
4. For the input mode, select **Check-in style sheet** and click **Next**.
5. For the XSLT Style Sheet filename, type the path to the XSLT style sheet or click **Browse**, locate the style sheet on your local disk, and click **Open**. The name should not have spaces or apostrophes in it.
6. Type comments in the **Check-in Comments** field.

   Note: Use the Check-in Comments field to note the purpose of the XSLT style sheet or explain the changes made to it.
7. Select the encoding that most closely matches the style sheet encoding and click **Next**.
8. If you do not want the XSLT style sheet to be enabled, click the **Enable for Business Processes** check box to clear it.
9. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
   - If Yes, click **Finish** to apply your changes.
   - If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking In an XSLT Style Sheet Using the Text Editor

You can also check in XSLT style sheets by typing or copying the content of an XSLT into the text editor.

To check an XSLT style sheet in to Application using the text editor:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Check-in**, click **Go**!
3. Type the name of the XSLT style sheet.
4. For the input mode, select **Style Sheet Text Editor** and click **Next**.
5. Type a description of the style sheet.

6. Under **XSL Style Sheet**, type or copy the content of the style sheet and click **Next**.

   Note: The text editor does not validate the style sheet.
7. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
   - If Yes, click **Finish** to apply your changes.
   - If No, click **Back** to make changes to your selections, or click Cancel to cancel without saving your changes.

Searching for an XSLT Style Sheet

To check in a new version, check out, enable, or disable an XSLT style sheet, you must first specify which one you want. You can locate an XSLT by name or from an alphabetic list.

Searching by name is more precise and provides fewer results. Searching from an alphabetical list shows all XSLT style sheets or ones beginning with a specified letter or digit.

To search for an XSLT style sheet by name:

1. From the **Deployment** menu, select **XSLT**.
2. Under **Search**, type the name of the XSLT style sheet. Case does not matter and you can type part of a name and click **Go**!

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

To search for an XSLT style sheet from a list:

1. From the Deployment menu, select XSLT.
2. Under List, select All or a specific letter or digit (0 - 9) and click Go!

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

About Search Results:

When you search for an XSLT style sheet, Application returns a results page. The results are displayed in a three-column table. Each row contains icons for the Source Manager and the Version Manager, the XSLT name, and XSLT type. You can sort the list alphabetically by name or type.

Source Manager:

The Source Manager enables you to check out an XSLT style sheet and check in a new version of that style sheet. It also displays the following information about an XSLT style sheet:

- Date that the XSLT style sheet was checked in
- Name of the user who checked in the XSLT style sheet
- Comments about changes that have been made

Version Manager:

The Version Manager enables you to enable or disable a version of an XSLT style sheet. If there are two or more versions, you can select a default.

The Version Manager also displays the following information about an XSLT style sheet and any of its versions:

- Which version is the default version
- Date that the XSLT style sheet version was checked in
- Name of the user who checked in the XSLT style sheet version
- Comments about changes that have been made

Editing an XSLT Style Sheet

After you have checked in a style sheet to Application, you can edit it without checking it out of Application.

To edit an XSLT style sheet in Application:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to edit. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet you want to edit, click **Source Manager**.
4. Next to the version you want to edit, click **Edit**.
5. Type a description of the changes you want to make to the style sheet.
6. Under **XSLT Style Sheet**, edit the style sheet as necessary and click **Next**.

   Note: The text editor does not validate the style sheet.
7. Select which version you want to be the default and click **Next**.
8. Review the settings for the XSLT style sheet. Are the settings correct?
   - If Yes, click **Finish** to apply your changes.
   - If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking In an Updated Version of an XSLT Style Sheet

If you update an XSLT style sheet that has been checked in to Application, you need to check in that style sheet again as an updated version.

To check an updated version of an XSLT style sheet in to Application:

1. From the **Deployment** menu, select XSLT.
2. Find the XSLT style sheet for which you want to check in a new version. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet for which you want to check in a new version, click **Source Manager**.
4. Next to **Check-in** a new version of this XSLT style sheet, click **Go**!
5. Type the path to the XSLT style sheet or click **Browse**, locate the XSLT style sheet, and click **Open**.
6. Type comments in the **Check-in comments** field and click **Next**. This field is required.

   Note: Use the Check-in comments field to note the purpose of the XSLT style sheet or explain the changes made to it.
7. Select the version you want to be the default and click **Next**.
8. If you do not want the XSLT style sheet to be enabled, click the **Enable for Business Processes** check box to clear it.

9. Review the settings for the XSLT style sheet you are checking in. Are the settings correct?
   - If Yes, click **Finish** to apply your changes. Application displays the message: *The system update has completed successfully*.
   - If No, click **Back** to make changes to your selections, or click **Cancel** to cancel without saving your changes.

Checking Out an XSLT Style Sheet

To edit an XSLT style sheet that has been checked in to Application and prevent anyone from modifying the file while you are making changes, you check out a version from Application. Checking out locks the source XSLT style sheet so that no one else can edit it while you are editing it. Use the Source Manager to check out a version of an XSLT style sheet.

To check out a version of an XSLT style sheet from Application:

1. From the **Deployment** menu, select XSLT.
2. Find the XSLT style sheet you want to check out. For more information, see *Searching for an XSLT Style Sheet*.
3. Next to the XSLT style sheet you want to check out, click **Source Manager**.
4. Next to the version you want to check out, select the encoding.

   Note: If a version has been checked in with an encoding other than the Application default of UTF-8, then you can check it out in UTF-8 or any other encoding that the style sheet has been checked in with.
5. Click **Check-out**.
6. Select **Save** then click **OK**. Application prompts you to choose a destination location. Browse to the location and click **OK** to save the file and complete checkout.

Enabling or Disabling an XSLT Style Sheet

Enabling a XSLT style sheet makes it available to the Application services and business processes.

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

- At the time you check it in to Application
- Through the Version Manager after the style sheet has been checked in

To enable or disable an XSLT style sheet with the Version Manager:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want to enable or disable.
3. Next to the XSLT style sheet you want to enable or disable, click **Version Manager**.
   - To enable an XSLT style sheet, click the empty **Enable** box and click **Save**. A check mark indicates the XSLT style sheet is enabled.
   - To disable an XSLT style sheet, click the checked **Enable** box and click **Save**. An empty box indicates the XSLT style sheet is disabled.

Specifying a Default XSLT Style Sheet

The default XSLT style sheet is the version that is available to business processes. One version must be selected as the default.

To specify a default XSLT style sheet:

1. From the **Deployment** menu, select **XSLT**.
2. Find the XSLT style sheet you want and click **Version Manager**.
3. Select the version you want to be the default and click **Save**.

Chapter 5. Build 5003 or Higher

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 Application platforms.
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	As part of a business process.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	None
Testing considerations	Use the Business Process Examples as a test.

How the Lock Service Works

The Lock service secures a business process and prevents other business processes from using the locked resources until the lock is released. The Lock service uses

the lock key and duration time that you set in the GPM for the business process to identify the lock to set and how long to keep the resources locked. You can lock all of the activities and services used in a business process by adding the Lock service to the beginning of a business process, after the Start and Sequence Start activities. Or, you can lock just some of the activities or services in a business process by adding the Lock service directly before the activities or services to be locked.

To release the lock, add another instance of the Lock service to the business process directly after the group of locked activities and services. If the entire business process was locked, add the second Lock service to the end of the business process, directly before the End Sequence and End activities.

Implementing the Lock Service

To implement the Lock service, complete the following tasks:

1. Create a configuration of the Lock service, or use the configuration installed with the Application, LockService. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Lock Service*.
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 the details of the currently 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.

Field	Description
DURATION	Time, in milliseconds, that the lock is applied. Required. The lock will time out or expire after this time. **Note:** If a business process halts due to an error, the service configuration will remain locked until you manually release the lock or restart the Application.
LOCK_KEY	The key for obtaining the lock. Required. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.
MAX_MEM_LOCK_PER_NODE_LISTED	Maximum number of local locks returned from each node. Valid value is any number greater then 0. Use this parameter if the value you specified in the ACTION field is List Locks. The default value is the value you specified in the defaultMaxMemLocksListedPerNode property in the centralops.properties.in file. Optional.
USER	User name associated with the lock (informational only). Optional. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.

Parameter(s) That Must be Added in BPML

The following additional parameter(s) can be used with Lock service by editing the BPML:

Parameter	Description
CLEAR_ON_START_UP	Clears the lock after you restart the Application. Use when you want to ensure that the lock is always clean when the Application is restarted. Optional. Valid values are true and false.
TimeStamp-MilliSeconds	Determines the time output in milliseconds since January 01, 1970, midnight.

Example Business Process 1

In this business process, a lock is applied by the Lock service. The lock is set to a duration of 600,000 milliseconds, and uses "Lock1" as the lock key. In this business process, the resource being locked is an instance of the Sleep service. Note that the sleep interval (duration) is 45 seconds. The Sleep service is followed by a second instance of the Lock service that releases the lock, thereby freeing up this configuration of the Sleep service configuration for other processes.

Note: The business process also includes onFault activities and messages in the event that one of the lock activities fails.

```
<process name="LockExample">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
      <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>
```

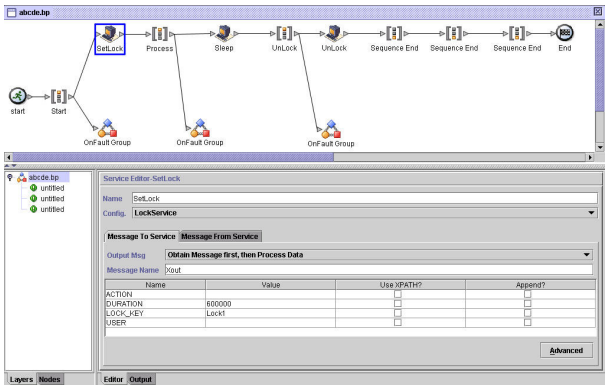
Example Business Process 2

In this business process, the time output is displayed in milliseconds since January 01, 1970, midnight.

```
<process name="LockExample">
  <rule name="Lock More than 2 hours old">
    <condition>number(/ProcessData/DateTime-MilliSeconds/text()) -
number(/ProcessData/debug_timestamp/text()) &gt; 7200000</condition>
  </rule>
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="LOCK_KEY" from="'LockTest'"></assign>
        <assign to="ACTION">LOCK</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="USER" from="'TestUser'"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
        <assign to="Status_Report" from="Status_Rpt('Report')"></assign>
      </input>
    </operation>

    <!-- Force the BP to fail -->
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="LOCK_KEY" from="'LockTest'"></assign>
        <assign to="ACTION">LOCK</assign>
        <assign to="DURATION">86400000</assign>
        <assign to="USER" from="'TestUser'"></assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="Xin">
        <assign to="." from="*"></assign>
        <assign to="Status_Report" from="Status_Rpt('Report')"></assign>
      </input>
    </operation>

    <onFault code="LOCK:Lock exists">
      <sequence name="Sequence Start">
        <operation name="ListLock">
          <participant name="LockService"/>
          <output message="Xout">
            <assign to="ACTION">LIST_LOCKS</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="Xin">
            <assign to="." from="DocToDOM(PrimaryDocument,'false')
/InMemoryLocks"></assign>
          </input>
        </operation>
        <operation name="Timestamp Utility">
          <participant name="TimestampUtilService"/>
          <output message="TimestampUtilServiceTypeInputMessage">
            <assign to="action">current_time</assign>
            <assign to="format">yyyyMMddHHmmssSSS</assign>
            <assign to="." from="*"></assign>
          </output>
          <input message="inmsg">
            <assign to="/ProcessData/DateTime" from="/inmsg/time/text()"
append="true"></assign>
            <assign to="/ProcessData/DateTime-MilliSeconds"
from="/inmsg/currentTimeMillis/text()" append="true"></assign>
          </input>
        </operation>

        <assign name="Assign" to="debug_timestamp"
from="/ProcessData/InMemoryLocks/NodeLocks/LockEntry[ResourceName/text() =
```

```
'LockTest']/TimeStamp/text()"></assign>
    <assign name="Assign" to="debug_timestamp_millisec"
from="/ProcessData/InMemoryLocks/NodeLocks/LockEntry[ResourceName/text() =
'LockTest']/TimeStamp-MilliSeconds/text()"></assign>

        <choice name="Choice Start">
          <select>
            <case ref="Lock More than 2 hours old" activity="SendErrorMessage"/>
          </select>
          <sequence name="SendErrorMessage">
            <assign to="Lock_Msg" append="true">Failed to obtain a lock!</assign>
          </sequence>
        </choice>
      </sequence>
    </onFault>
  </sequence>
</process>
```

The following GPM screen shows the example business process graphically. Note that the instance of the Lock service called SetLock is selected and its properties are displayed in the Service Editor in the lower half of the screen:



Lock Service - Frequently Asked Questions

How do I determine what the duration of a lock should be?

- Set the lock duration value carefully (generally 2-3 times what the estimated time of execution will be) so that lock does not time out before the business process reaches the unlock step.

If the lock or unlock step fails (or any step in the business process between the two Lock services), are there any "cleanup" activities that need to be done?

- Halted business processes can be terminated, or they can be left in halted state until the issue is resolved, and then restarted or resumed (as appropriate for the persistence level). The path to take depends on the needs of your business with regard to the business process itself. For example, is it mission-critical? Do other processes depend on its completion? Is the locked resource going to cause other business processes that use it to halt? Was the error caused by a problem in the configuration of the locked resource? Will this need to be corrected before using it again?

There are two general steps to follow first:

1. Check the lock manager page to see what's locked.
2. Check **Business Process** > **BP Monitor** > **Current Processes** for more information about the error.

Then, once you have determined what caused the error, you can decide when to release the lock and when to terminate, restart, or resume the business process.

To manually release a locked resource:

Go to **Operations** > **Lock Manager**, and click **Go**! in the List panel. The locked resources are displayed on a results page. Locate the resources from your business process that are locked and clear the Lock checkbox for the resources.

To terminate, restart, or resume a business process:

Go to **Business Process** > **Monitor** > **Current Processes** and select the ID of the halted instance of your business process. From the page displayed, you can select the appropriate action for this business process: terminate, restart, or resume.

Are there any best practices for using the Lock service in a business process?

Use the Lock service twice in a business process-one to lock resources and one to unlock them. The first instance precedes the resources to be locked and the second instance follows them. See the *Business Process Examples* 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.

Chapter 6. Build 5002 or Higher

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 Application platforms.
Related services	None
Application requirements	None
Initiates business processes?	No
Invocation	As part of a business process.
Business process context considerations	None
Returned status values	Success, Error
Restrictions	None
Testing considerations	Use the Business Process Examples as a test.

How the Lock Service Works

The Lock service secures a business process and prevents other business processes from using the locked resources until the lock is released. The Lock service uses

the lock key and duration time that you set in the GPM for the business process to identify the lock to set and how long to keep the resources locked. You can lock all of the activities and services used in a business process by adding the Lock service to the beginning of a business process, after the Start and Sequence Start activities. Or, you can lock just some of the activities or services in a business process by adding the Lock service directly before the activities or services to be locked.

To release the lock, add another instance of the Lock service to the business process directly after the group of locked activities and services. If the entire business process was locked, add the second Lock service to the end of the business process, directly before the End Sequence and End activities.

Implementing the Lock Service

To implement the Lock service, complete the following tasks:

1. Create a configuration of the Lock service, or use the configuration installed with the Application, LockService. See *Managing Services and Adapters*. For information about the fields specific to this service, see *Configuring the Lock Service*.
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 the details of the currently 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.

Field	Description
DURATION	Time, in milliseconds, that the lock is applied. Required. The lock will time out or expire after this time. **Note:** If a business process halts due to an error, the service configuration will remain locked until you manually release the lock or restart the Application.
LOCK_KEY	The key for obtaining the lock. Required. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.
MAX_MEM_LOCK_PER_NODE_LISTED	Maximum number of local locks returned from each node. Valid value is any number greater then 0. Use this parameter if the value you specified in the ACTION field is List Locks. The default value is the value you specified in the defaultMaxMemLocksListedPerNode property in the centralops.properties.in file. Optional.
USER	User name associated with the lock (informational only). Optional. If using two instances of the Lock service in the same business process (one to lock, the other to unlock), this key must be the same value for both.

Parameter(s) That Must be Added in BPML

The following additional parameter(s) can be used with Lock service by editing the BPML:

Parameter	Description
CLEAR_ON_START_UP	Clears the lock after you restart the Application. Use when you want to ensure that the lock is always clean when the Application is restarted. Optional. Valid values are true and false.

Business Process Example

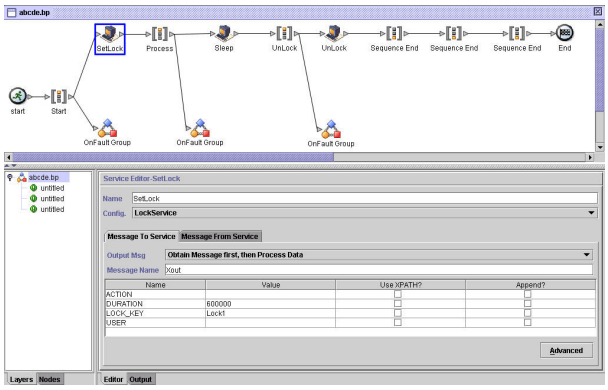
The following example illustrates how the Lock service could be used in a business process.

In this business process, a lock is applied by the Lock service. The lock is set to a duration of 600,000 milliseconds, and uses "Lock1" as the lock key. In this business process, the resource being locked is an instance of the Sleep service. Note that the sleep interval (duration) is 45 seconds. The Sleep service is followed by a second instance of the Lock service that releases the lock, thereby freeing up this configuration of the Sleep service configuration for other processes.

Note: The business process also includes onFault activities and messages in the event that one of the lock activities fails.

```
<process name="LockExample">
  <sequence name="Start">
    <operation name="SetLock">
      <participant name="LockService"/>
      <output message="Xout">
        <assign to="DURATION">600000</assign>
        <assign to="LOCK_KEY">Lock1</assign>
        <assign to="CLEAR_ON_START_UP">true</assign>
        <assign to="." from="*"></assign>
      </output>
      <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 Examples* 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.

Chapter 7. Build 5001 or Higher

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 Application.
Platform availability	All supported Application 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	• buildResponse - If an exception is thrown during build process, the "exception-message" node is returned to ProcessData with the exception message. • parseResponse - If an exception is thrown during parse process, the "exception-message" node is returned to ProcessData with the exception message.
Restrictions	None

Service name	Cryptographic Message Service
Testing considerations	• You should use the right certificates for signing or encryption/decryption. • If you receive an error with the condition that certificates used for signing or decrypting are not created with a storepass value of integrator and are created with a keypass value of integrator, see your system administrator.

How the Cryptographic Message Service Works

Cryptographic Message Service (CMS) builds and parses secure messages in Secure MIME (SMIME), Distinguished Encoding Rules (DER), or Privacy Enhanced Email (PEM) format.

The security features of CMS are digital signature and encryption. The Digital signature feature provides authentication, message integrity, and non-denial with proof 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 Application Admin Console and in the GPM as necessary. For information, see *Configuring the Cryptographic Message Service*.
3. Use the Cryptographic Message service in a business process.

System Administrator Tasks

The following procedures describe the system administrator tasks for cryptographic message service.

Importing a keyCert into Application

1. Login to Application.
2. Select **Trading Partner** -> **Digital Certificates** -> **System**.
3. Select **Key Certificate** under Check in.
4. Enter the Certificate Name and Private Key Password.
5. Select the certificate and assign an alias to it.

6. Review and click **Finish**. You can use this certificate in your BPML associated with the appropriate field (signingCert or decryptCert).

Importing a Public Certificate into Application

1. Login to Application.
2. Select **Trading Partner** -> **Digital Certificates** -> **Trusted**.
3. Select **New Certificate** under Check in.
4. Select the certificate and click **Next**.
5. Enter the Certificate Name and click **Next**.
6. Review and click **Finish**. 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 Application 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: • None - You do not want to include this configuration in a group at this time (default). • Create New Group - You can enter a name for a new group in this field, which is then created along with this configuration. • Select Group - If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. For more information about service groups see *Managing Services and Adapters*.

Field	Description
Cryptographic Message Syntax	Drop-down menu containing a list of cryptographic message syntaxes for building cryptographic messages. Required. Valid values are: • CMS (default) • PKCS#7
Security Type	Drop-down menu containing the security type for building cryptographic messages. Required. Valid values are: • Encrypted Only (default) - Encrypts the message only. • Detached Signed Only - Signs the original document and leaves the signature detached from the original document. If the message output format is SMIME, multipart MIME message will separate the original document and signature. If the message output format is DER or PEM, only detached signature will be returned by the service. • Embedded Signed Only - Signs the original document and embeds the original document inside the signature. • Detached Signed and Encrypted - Creates detached signed signature and encrypts the signed message. If the message output format is SMIME, the encryption is applied on the multipart MIME message. If the message output format is DER or PEM, the encryption is applied on the detached signature only. • Embedded Signed and Encrypted - Creates embedded signed signature and encrypts the signed message.
Message Output Format	Message output format for generating the signed or encrypted message. Required. Valid values are: • SMIME (default) - The signed or encrypted message will be output in MIME format. • DER - The signed or encrypted message will be output in DER encoded format. • PEM - The signed or encrypted message will be output in PEM encoded format, which is a base64 encoded DER format and enclosed between a start and an end boundary.

Field	Description
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: • application (default) • text • message • image • video • audio
Document MIME Sub Content Type	This parameter is enabled only if you select SMIME as the message output format. MIME sub content type for the document that needs to be packages. If the input document is set with the sub content type, the value will override the setting here. Optional. Valid values are: • octet-stream (default) • plain • edi-x12 • edifact • edi-consent • 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: • Base64 (default) • 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: • Yes (default) • No

Field	Description
Encryption Algorithm:	Content encryption algorithm. Optional. Valid values are: • Triple DES (3DES) 168 CBC with PKCS5 padding (default) • 56-bit DES CBC with PKCS5 padding • 128-bit RC2 CBC with PKCS5 padding • 40-bit RC2 CBC with PKCS5 padding • 128-bit AES CBC with PKCS5 padding • 192-bit AES CBC with PKCS5 padding • 256-bit AES CBC with PKCS5 padding
Encryption Certificate(s)	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: • Single Signature (default) • Multiple Signatures • Counter Signature • No Signature Required
Signing Algorithm	The signing algorithm to hash the document. Optional. Valid values are: • SHA1 (default) • MD5
Signing Certificate(s)	Private certificates to sign the document. Optional. Valid values are: • Select a signing certificate if you have selected Single Signature. • Select a list of signing certificates for multiple users to sign the document if you have selected Multiple Signatures. • Select a list of signing certificates for multiple users to sign the document and countersign the signature if you have selected Counter Signature.
Message Input Format	Message input format for parsing the signed or encrypted message. Required. Valid values are: • SMIME (default) • DER • PEM

Field	Description
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: • Encrypted Only (default) - The inbound message is encrypted only. • Detached Signed Only - The inbound message is signed in detached format. • Embedded Signed Only - The inbound message is signed in embedded format. • Detached Signed and Encrypted - The inbound message is signed in detached format and then encrypted. • Embedded Signed and Encrypted - The inbound message is signed in embedded format and then encrypted.
Decryption Certificate	Private certificate to decrypt the cryptographic message. Optional.
Signature Verification Certificate(s)	Public certificates to verify signed cryptographic message. Optional. **Note:** You can select single certificate if the inbound message is signed by one certificate or select a list of certificates if multiple certificates sign the inbound message. Based on the certificates list sequence, counter signature verification starts from the first level of the signature.

Parameters That Must Be Added in BPML

The following additional parameters are available for use with the Cryptographic Message service, but can only be added by editing your business process manually. This parameter is not available through the Admin console or the GPM:

Parameter	Description
Action	The two values are either *build* or *parse*. Required.
pipelineTimeout	Controls the duration of building or parsing process. By default, the value is 300 seconds and can be increased to process large files. Optional.

Business Process Examples

The parameters passed from the BPML precede over the parameters passed from the service. The following BPML examples illustrate using the cryptographic message service instance:

Example Business Process 1

The following BPML builds the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```
<process name="cryptomsg_build">
   <sequence>
       <operation name="Crypto Message Service">
           <participant name="CryptoMsgService"/>
           <output message="buildRequest">
               <assign to="." from="*"/>
               <assign to="action">build</assign>
           </output>
           <input message="buildResponse">
               <assign to="." from="*"/>
           </input>
       </operation>
    </sequence>
</process>
```

Example Business Process 2

The following BPML parses the cryptographic messages based on the parameters passed from BPML to the service or the configuration set in CMS instance configuration.

```
<process name="cryptomsg_parse">
   <sequence>
       <operation name="Crypto Message Service">
           <participant name="CryptoMsgService"/>
           <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">false</assign>
                <assign to="bootstrap">false</assign>
                <assign to="deleteAfterCollect">false</assign>
                <assign to="." from="*"/>
            </output>
            <input message="FileSystemOutputMessage">
                <assign to="." from="*"/>
            </input>
          </operation>
           <assign to="detachedDoc" from="PrimaryDocument/@SCIObjectID"/>
          <operation name="Import Document Request">
            <participant name="CryptoMsgTestFSA"/>
            <output message="FileSystemInputMessage">
                <assign to="Action">FS_COLLECT</assign>
                <assign to="collectionFolder" from="'/gisinstall'"/>
                <assign to="filter" from="'signed_msg.txt'"/>
  <assign to="useSubFolders">false</assign>
                <assign to="bootstrap">false</assign>
                <assign to="deleteAfterCollect">false</assign>
                <assign to="." from="*"/>
            </output>
            <input message="FileSystemOutputMessage">
```

```
                <assign to="." from="*"/>
            </input>
        </operation>
        <operation name="Crypto Message Service">
           <participant name="CryptoMsgService"/>
           <output message="parseRequest">
               <assign to="." from="*"/>
               <assign to="action">parse</assign>
               <!--securityType=3 Encrypted Only,
 securityType=1 Detached Signed Only,
 securityType=2 Embedded Signed Only,
           securityType=4 Detached Signed and Encrypted,
              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.	`<EncryptCerts>`
	`  <Cert1>`
	`  <Name>smime_pub1</Name>`
	`  <Status>ok</Status>`
	`  <ExpiryTime>20350726074016Z</ExpiryTime>`
	`  </Cert1>`
	`  <Cert2>`
	`  <Name>smime_pub2</Name>`
	`  <Status>ok</Status>`
	`  <ExpiryTime>20350726074056Z</ExpiryTime>`
	`  </Cert2>`
	`</EncryptCerts>`

Scenario	Output
Certificate used for encryption or signing has expired.	<SigningCerts> <Cert1> <Name>smime_pub1</Name> <Status>expired</Status> <ExpiryTime>20070726074016Z</ExpiryTime> </Cert1> </SigningCerts> <exception-message>xxx</exception-message> Or <EncryptCerts> <Cert1> <Name>smime_pub1</Name> <Status>expired</Status> <ExpiryTime>20070726074016Z</ExpiryTime> </Cert1> <Cert2> <Name>smime_pub2</Name> <Status>ok</Status> <ExpiryTime>20350726074056Z</ExpiryTime> </Cert2> </EncryptCerts>
Certificate used for encryption has been revoked.	<EncryptCerts> <Cert1> <Name>cert1</Name> <Status>revoked</Status> </Cert1> </EncryptCerts> <exception-message>xxx</exception-message>
Certificate used for encryption fails to process. For example, if the encryption certificate is not found in Application.	<EncryptCerts> <Cert1> <Name>cert1</Name> <Status>error</Status> </Cert1> </EncryptCerts> <exception-message>xxx</exception-message>

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	<DecryptionResult> <DecryptionCertName>smime_priv1</DecryptionCertName> <DecryptionCertStatus>ok</DecryptionCertStatus> <DecryptionCertExpiryTime>20350726074016Z </DecryptionCertExpiryTime> <Status>passed</Status> </DecryptionResult>
Decryption certificate not found in Application	<DecryptionResult> <DecryptionCertName>cert1</DecryptionCertName> <DecryptionCertStatus>error</DecryptionCertStatus> <Status>failed</Status> </DecryptionResult>

Scenario	Output
Decryption certificate failed to decrypt	<DecryptionResult> <DecryptionCertName>smime_priv2</DecryptionCertName> <DecryptionCertStatus>ok</DecryptionCertStatus> <DecryptionCertExpiryTime>20350726074056Z </DecryptionCertExpiryTime> <Status>failed</Status> </DecryptionResult>
Signature verification is passed	<SignatureVerificationResults> <SignatureVerificationResult1> <VerificationCertName>smime_dsa_pub</VerificationCertName> <VerificationCertStatus>ok</VerificationCertStatus> <VerificationCertExpiryTime>20350812084354Z </VerificationCertExpiryTime> <SigningTime>20080917021420Z</SigningTime> <Status>passed</Status> </SignatureVerificationResult1> <SignatureVerificationResult2> <VerificationCertName>smime_pub4</VerificationCertName> <VerificationCertStatus>ok</VerificationCertStatus> <VerificationCertExpiryTime>20350726074148Z </VerificationCertExpiryTime> <SigningTime>20080917021420Z</SigningTime> <Status>passed</Status> </SignatureVerificationResult2> <Status>passed</Status> </SignatureVerificationResults>
Signature verification fails	<SignatureVerificationResults> <SignatureVerificationResult1> <VerificationCertName>smime_pub4</VerificationCertName> <VerificationCertStatus>ok</VerificationCertStatus> <VerificationCertExpiryTime>20350726074148Z </VerificationCertExpiryTime> <SigningTime>20080917021549Z</SigningTime> <Status>passed</Status> </SignatureVerificationResult1> <SignatureVerificationResult2> <VerificationCertName>smime_pub3</VerificationCertName> <VerificationCertStatus>ok</VerificationCertStatus> <VerificationCertExpiryTime>20350726074122Z </VerificationCertExpiryTime> <SigningTime>20080917021549Z</SigningTime> <Status>failed</Status> </SignatureVerificationResult2> <Status>failed</Status> </SignatureVerificationResults>
Multiple signature verification fails	<SignatureVerificationResults> <SignatureVerificationResult1> <SigningTime>20080917071327Z</SigningTime> <Status>nomatched_verificationCert</Status> </SignatureVerificationResult1> <SignatureVerificationResult2> <VerificationCertName>smime_pub3</VerificationCertName> <VerificationCertStatus>ok</VerificationCertStatus> <VerificationCertExpiryTime>20350726074122Z </VerificationCertExpiryTime> <SigningTime>20080917021549Z</SigningTime> <Status>failed</Status> </SignatureVerificationResult2> <Status>failed</Status> </SignatureVerificationResults>

Scenario	Output
Signature verification certificate is revoked	<SignatureVerificationResults> <SignatureVerificationResult1> <SigningTime>20080917024531Z</SigningTime> <VerificationCertName>serenaCRL1</VerificationCertName> <VerificationCertStatus>revoked</VerificationCertStatus> </SignatureVerificationResult1> <Status>failed</Status> </SignatureVerificationResults>

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.

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.

System name	JMS Queue Adapter
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.
Select a Group	Select one of the options: • None - You do not want to include this configuration in a group at this time. • Create New Group - You can enter a name for a new group in this field, which will then be created along with this configuration. • Select Group - If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. **Note:** See *Using Service Groups*.
Connection Type	Whether or not the adapter uses JNDI lookup to connect to the remote JMS Queue server. Valid values are: • Using Jndi - Uses JNDI lookup. • 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.

Field	Description
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: • Yes - Debug messages will be logged. • No - Debug messages will not be logged.
Queue Type (Action)	Type of queue to access. Required. Valid values are: • Queue Send - Send messages. • 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). • 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.

Field	Description
Message Type (Payload)	Type of message to send. Used only if queue type is Queue Send. Valid values are: • BytesMessage • ObjectMessage • StreamMessage • TextMessage
Bootstrap Workflow (InitialWorkFlowId)	Business process to initiate when data is received. Used only if queue type is Queue Receive Async. Required.
Document Storage Type (docStorageType)	Defines how the document will be stored in the system. Used only if queue type is Queue Receive Async. Required. Valid values: • System Default • Database • File System **Note:** See *Selecting a Document Storage Method for Bootstrap Adapters*.
Bootstrap Mode (BootstrapMode)	The mode where the business process is started and executed. Valid values: • AsyncBootstrap - Mode that provides default Sterling Integrator functionality. The business processes started by the adapter are placed on the Sterling Integrator queues and executed asynchronously to the adapter. • FifoBootstrap - Mode that executes business processes in First-In, First-Out order. See *FIFO Message Processing Enhancement for Sterling Integrator 5.0* for additional information about this processing mode. • NonQueuedBootstrap- Mode executes business processes within the adapter's thread of execution. This provides low latency execution but restricts the adapter to a single thread of execution. See *Business Processing Queued and Non-Queued Processing* for additional information.
FIFO Initialization Business Process (FifoInitializationBpName)	Specify the name of the business process that will be executed to determine FIFO routing key. **Note:** This option is only available when the adapter is in FIFO bootstrap mode.

Field	Description
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. **Note:** This option is available only for the Async bootstrap mode. FIFO and Non-Queued bootstrap modes make use of a single bootstrap thread per adapter.
Buffer Size (BufferSize)	Size of the buffer when receiving data. Used only if queue type is Queue Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional.
Document Filename (OutputFileName)	If you choose Queue Receive Async as the queue type for the JMS Queue adapter, then you can specify a file name for the data that the JMS Queue receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form `<nodename>_yyyymmddhhmmssiii`.
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.
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:

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/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/+` as shown in the example below.

Example of ProcessData after a batch receive was performed:

```
<ProcessData>
  <JMS>
    <DocumentCount>3</DocumentCount>
    <Document1 SCIObjectID="1833955:1060de6d03d:-697b">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156360801</messageID>
      <priority>4</priority>
      <timestamp>1125515636080</timestamp>
    </Document1>
    <Document2 SCIObjectID="1833955:1060de6d03d:-6978">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
      <expiration>0</expiration>
      <messageID>ID:234-11255156361102</messageID>
      <priority>4</priority>
      <timestamp>1125515636110</timestamp>
    </Document2>
    <Document3 SCIObjectID="1833955:1060de6d03d:-6975">
      <redelivered>false</redelivered>
      <deliveryMode>2</deliveryMode>
      <destination>testqueue</destination>
```

```
      <expiration>0</expiration>
      <messageID>ID:234-11255156361243</messageID>
      <priority>4</priority>
      <timestamp>1125515636124</timestamp>
    </Document3>
  </JMS>
</ProcessData>
```

JMS Topic Adapter

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.

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.
Select a Group	Select one of the options: • None - You do not want to include this configuration in a group at this time. • Create New Group - You can enter a name for a new group in this field, which will then be created along with this configuration. • Select Group - If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. **Note:** See *Using Service Groups*.
Connection Type	Whether or not the adapter uses JNDI lookup to connect to the remote JMS Topic server. Valid values are: • Using Jndi - Uses JNDI lookup. • 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.

Field	Description
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: • Yes - Debug messages will be logged. • No - Debug messages will not be logged.
Topic Type (Action)	Type of topic to access. Valid values are: • Topic Send - Sends messages. • 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). • 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: • BytesMessage • ObjectMessage • StreamMessage • TextMessage
Bootstrap Workflow (InitialWorkFlowId)	Business process to initiate when data is received. Used only if topic type is Topic Receive Async. Required.
Document Storage Type (docStorageType)	Defines how the document will be stored in the system. Used only if topic type is Topic Receive Async. Required. Valid values: • System Default • Database • File System **Note:** See *Selecting a Document Storage Method for Bootstrap Adapters*.

Field	Description
Bootstrap Mode (BootstrapMode)	The mode where the business process is started and executed. Valid values: • AsyncBootstrap - Mode that provides default Sterling Integrator functionality. The business processes started by the adapter are placed on the Sterling Integrator queues and executed asynchronously to the adapter. • FifoBootstrap - Mode that executes business processes in First-In, First-Out order. See *FIFO Message Processing Enhancement for Sterling Integrator 5.0* for additional information about this processing mode. • NonQueuedBootstrap- Mode executes business processes within the adapter's thread of execution. This provides low latency execution but restricts the adapter to a single thread of execution. See *Business Processing Queued and Non-Queued Processing* for additional information.
FIFO Initialization Business Process (FifoInitializationBpName)	Specify the name of the business process that will be executed to determine FIFO routing key. **Note:** This option is only available when the adapter is in FIFO bootstrap mode.
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. **Note:** This option is available only for the Async bootstrap mode. FIFO and Non-Queued bootstrap modes make use of a single bootstrap thread per adapter.
Buffer Size (BufferSize)	Size of the buffer when receiving data. Used only if topic type is Topic Receive Async. Enables you to fine-tune the performance of the adapter according to data expectations. Default is 30000. Optional.
Document Filename (OutputFileName)	If you choose Topic Receive Async as the topic type for the JMS Topic adapter, then you can specify a file name for the data that the JMS Topic receives. A unique file name generator placeholder, %^, can be used to generate a sequence in the form *nodename_yyyymmddhhmmsslll*.
Connection retry attempts (RetryCount)	Maximum number of connection retry attempts. Used only if topic type is Topic Receive Async. Specify -1 for an infinite number of retry attempts. Default is 20. Optional.

Field	Description
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:

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>
```

MSMQ Adapter

The following table provides an overview of the MSMQ adapter:

System name	MSMQ
Graphical Process Modeler (GPM) categories	All Services, Custom, Sync Mode, Transactional Mode **Note:** This adapter will not display in the GPM stencils until you create a configuration of the adapter in the Sterling Integrator Admin UI.

System name	MSMQ
Description	Sends messages to and reads messages from a remote Microsoft® Message Queue. You can configure the MSMQ adapter to send and retrieve messages: • Send mode – Sends a message from a business process to a named queue on a specified Microsoft Windows® MSMQ server. The MSMQ adapter also returns pertinent message metadata to the business process after sending the message. • Retrieve mode – In a business process, the MSMQ adapter retrieves messages from a specified queue on a specified MSMQ server. When in retrieve mode, the MSMQ adapter continues to retrieve messages until the queue is empty. The MSMQ adapter also runs a specified business process for each message in the queue and passes the message body and selected metadata parameters to the business process that is running.
Business Usage	Processing messages from an MSMQ server as part of a business process.
Preconfigured?	No
Requires third party files?	Yes. Requires Jacob.lib (Com-java bridge package). This is included in the msmqbundle_4_1_0.jar, which is included with Sterling Integrator. See *Implementing the MSMQ Adapter*.
Platform availability	All supported Sterling Integrator platforms.
Related services	No
Application Requirements	The MSMQ adapter supports MSMQ version 1.1 and later. The MSMQ adapter supports 40-bit encryption for versions 1.1 and later and 128-bit encryption in versions 2.0 and later.
Initiates business processes?	Yes, when using the RETRIEVE action (reading a message from the queue).
Invocation	Runs as a service within a business process.
Returned status values	Success – The run was successful for the specified action. Error – An error occurred when the specified action was executed.

System name	MSMQ
Restrictions	Restrictions: • The MSMQ adapter supports single message transactions for guaranteed exactly-once delivery of messages. Other transaction modes are not supported. • The MSMQ adapter supports external transactions to MSMQ; that is, a transaction from Sterling Integrator for a SEND action. This only applies to transactional queues in MSMQ. • The MSMQ adapter does not support MSMQ acknowledgements.
Persistence Level	Full (default). **Note:** When running in transactional mode, this setting is overridden at the business process level and set to Zero persistence.

Requirements

To use the MSMQ adapter, you need:

- The connection from Sterling Integrator to the system where the MSMQ server resides.
- The port number of the system where the MSMQ server resides. This port number is used by the MSMQ adapter.
- The queue must be preconfigured in the MSMQ server. This queue name is used by the MSMQ adapter. The queue can be transactional or non-transactional, depending on how you need to use the MSMQ adapter.

How the MSMQ Adapter Works

The following steps summarize how the MSMQ adapter works in a business process within Sterling Integrator:

1. The MSMQ adapter extracts the message body from the incoming business process context and sends it to a remote system Microsoft Message Queue.
2. The MSMQ adapter creates a correlation ID and returns it in the process data elements.
3. The adapter retrieves any response from the Microsoft Message Queue and places it back into the business process context.
4. The adapter passes the updated business process context back to the business process and places any MSMQ elements into process data.

The following table contains the MSMQ elements placed into the process data after the MSMQ adapter runs:

MSMQ Element	Description
MSMQ_ARRIVEDTIME	When the messaged arrived at the queue.
MSMQ_BODY_TYPE	Whether the message is a string or byte array.
MSMQ_CORRELATIONID	ID number used to correlate related messages.

MSMQ Element	Description
MSMQ_EXTENSION	A place to put additional information associated with the message.
MSMQ_FIRST_IN_XACT	Whether the message was the first message in a transaction.
MSMQ_LAST_IN_XACT	Whether the message was the last message in a transaction.
MSMQ_XACTID	Message ID from a transaction queue.
MSMQ_LABEL	Label for the message.
MSMQ_MSGID	Message ID from a regular queue.
MSMQ_PRIORITY	Message priority level.
MSMQ_RETURNED_MESSAGE	Retrieved message.
MSMQ_XACTIONAL QUEUE	Name of the transactional queue from which the message was received.

Implementing the MSMQ Adapter

To implement the MSMQ adapter, complete the following tasks:

1. Activate your license for the MSMQ adapter. See *An Overview of Implementing Services*.
2. Install the MSMQPrime component.
3. Create and configure an MSMQ adapter configuration. For general instructions on creating a service configuration, see *Creating a Service Configuration*. For descriptions of the MSMQ adapter configuration parameters, see *Configuring the MSMQ Adapter*.
4. Use the MSMQ adapter in a business process.

Installing MSMQPrime

MSMQPrime is a component of the MSMQ adapter and shoud be deployed within the same Microsoft network where the MSMQ server resides. It can be co-located with the MSMQ server, the Application server, or an independent server. It must be able to reach the MSMQ server queue utilizing the Microsoft resource naming convention. The MSMQ adapter connects to the MSMQPrime component, which in turn, performs the send and receive actions and interacts with the MSMQ server. MSMQPrime listens on the port that is part of the MSMQ adapter configuration.

To install MSMQPrime, complete the following steps:

1. Locate msmqbundle_xxx.jar in the Sterling Integrator under `<INSTALL_DIR>`/client/msmq folder.
2. On the Windows MSMQ server host, create a folder for MSMQPrime. For example, C:\MSMQ.
3. Copy msmqbundle_xxx.jar to the folder you just created.
4. Change directory to that folder, and use winzip to unbundle the .jar file.
5. Copy all of the files in the InstallJavaService folder to the MSMQ folder so that installwindowsservice, msmqproperties, and the Jacob folder are in the same directory.
6. Copy Jacob.dll from the msmqbundle*.jar/Jacob/1_7 folder to the C:\WINDOWS\system32 folder.

7. Install the Java jdk version 1.5.0_11. Note the installation path.
8. Modify start_msmqPrime.cmd to use the folder you created in step 2. Set the MSMQADAPTER parameter to the folder you created in step 2. Set the JAVA parameter to point to the bin directory in the Java path created in step 5; that is, C:\jdk1.5.0_11\bin.

   Note: If Java is installed in the default installation folder in C:\Program Files\Java\jdk1.5.0_11, you have to reference it as C:\Progra~1\Java\jdk1.5.0_11\bin.
9. Change `MSMQ_SERVER_PORT` in msmqprime.properties, if necessary. The default is 8085. This is the port msmqPrime will run on.
10. Run start_msmqPrime.cmd. This script should be run by the user who has permission to create queues, read, and send messages to the MSMQ server. This process must be running continually if your MSMQ adapter needs to access it. It is recommended to convert it to an automatically started Windows service.
11. Create a configuration of the MSMQ adapter in your Sterling Integrator and configure it to point to this msmqPrime.
12. Configure a service instance of MSMQ adapter in the Sterling Integrator to point to this msmqPrime.
13. Verify that the MSMQ adapter configuration is talking to this msmqPrime by including it in a busi-ness process and running it.
14. If desired for testing purposes, turn on debug mode in msmqPrime with the following command by passing -debug as an argument to MSMQPrimaImpl in start_msmq.cmd.

    The debugon option generates detailed logs.

    Note: The msmqbundle_xxx.jar you use to create the MSMQPrime component must be from the same Sterling Integrator installation as the MSMQ adapter that it will talk to. You need to redeploy the msmqbundle_xxx.jar to the Windows MSMQ server host when an Sterling Integrator patch is installed. This ensures the new code changes are synchronized with MSMQPrime.

Configuring the MSMQ Adapter

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

Sterling Integrator Configuration

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

Note: This table contains configuration parameters for both Send and Retrieve. The field names in parentheses represent the corresponding field names in the GPM. This information is provided for your reference.

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

Field	Description
Select a Group	Select one of the options: • None – You do not want to include this configuration in a group at this time. • Create New Group – You can enter a name for a new group in this field, which will then be created along with this configuration. • Select Group – If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list.
Hostname(HostName)	Host name of the MSMQ server with which this configuration of the adapter communicates. Send and Retrieve parameters. Required.
PortNumber(PortNumber)	The port number of the MSMQ server host. Required.
Queue Path Name(QueuePathName)	Typical queue path name in the form `machineName\queueName`. Send and Retrieve parameters. Required.
EX_TRANSACTION	To include send action in the Sterling Integrator engine transaction. Valid values are TRUE and FALSE (default). Send parameter. Optional.

GPM Configuration

The following table describes the fields used to configure both Send and Retrieve configurations for the MSMQ adapter in the GPM:

Field	Description
Config	Name of the adapter configuration.
Action	Values are Send and Retrieve. Required if the action is Retrieve. If the action is Retrieve, specify business process. If the Action is Send, specify body type.
BodyType	Valid values are String (default) and Byte array. Send parameter. Optional.
BusinessProcessName	Business process that is started upon receipt of a message from the queue. Retrieve parameter. Required.
Delivery	Valid values are Recoverable (default) and Express. Send parameter. Optional.
EncryptionAlgorithm	CALG_RC2 (default) or CALG_RC4. Send parameter. Optional.
MaxTimeToReachQueue	Maximum time allowed for a message to reach its destination. Send parameter. Default is 300 seconds. Optional.

Field	Description
MaxTimeToReceive	Maximum time, in seconds, for a message to be received before it is discarded from the queue. Send parameter. Default is 0. Optional. **Note:** If set to 0, the message is not discarded.
MessageLabel	Label for message being sent to queue. Optional.
MessagePriority	Valid values are 0 - 7, where 7 represents the highest priority messages in the queue. The highest priority messages are received first. Send parameter. Default is 3. Optional. **Note:** This parameter setting does not apply to messages sent to a transactional queue, which will always have a message priority of 0.
PrivacyLevel	Used to request encryption. Valid values are None (default), Base (40-bit) and Enhanced (128-bit, supported in MSMQ version 2.0 and later). Send parameter. Optional.

Business Process Examples

The following examples illustrate using the MSMQ adapter for send and receive actions:

Send Action

```
<process name = "MSMQ_Base">
  <sequence name="Test Sequence">
   <operation name="MSMQ Commn">
      <participant name="MSMQAdapter"/>
      <output message="outmsg">
       <assign to="Action">SEND</assign>
       <assign to="PortNumber">0000</assign>
       <assign to="QueuePathName">server1\testqueue</assign>
       <assign to="MaxTimeToReceive">3600</assign>
       <assign to="MaxTimeToReachQueue">3600</assign>
       <assign to="MessagePriority">3</assign>
       <assign to="Delivery">RECOVERABLE</assign>
       <assign to="PrivacyLevel">Base</assign>
       <assign to="MessageLabel">Base</assign>
       <assign to="BodyType">BYTE ARRAY</assign>
       <assign to="GIS_TRANSACTION">TRUE</assign>
       <assign to="." from="*" />
     </output>
      <input message="inmsg">
       <assign to="." from="*" />
 </input>
    </operation>
 </sequence>
</process>
```

Retrieve Action

```
<process name = "MSMQ_Receive">
  <sequence name="Test Sequence">
    <operation name="MSMQ Commn">
     <participant name="MSMQAdapter"/>
     <output message="outmsg">
```

```
        <assign to="Action" from="'RETRIEVE'" />
        <assign to="PortNumber" from="'0000'" />
        <assign to="QueuePathName" from="'server1\testqueue'" />
       <assign to="MessageLabel" from="'Receive'" />
       <assign to="BusinessProcessName" from="'MSMQ_FileSystem'" />
        <assign to="." from="*" />
     </output>
      <input message="inmsg">
       <assign to="." from="*" />
 </input>
    </operation>
 </sequence>
</process>
```

PGP Package Service

Pretty Good Privacy (PGP) is an open standard data encryption and decryption tool. The PGP Package service, in conjunction with the PGP Server Manager, enables you to encrypt and digitally sign documents using PGP.

The following table provides an overview of the PGP Package service:

System name	**PGP Package service**
Graphical Process Modeler (GPM) category	All Services
Description	This service encrypts and digitally signs a document based on the Open PGP standard, using public key or conventional cryptography.
Business usage	Use this service to encrypt and sign a document in the document area of process data.
Usage example	A business process is executed to encrypt and sign a document, based on the information stored in a PGP profile.
Preconfigured?	Yes. A configuration called PGP Package Service is installed with Application.
Requires third-party files?	No

System name	PGP Package service
Platform availability	All supported Application platforms, with the following restrictions: **For NAI McAfee eBusiness Server 8.1** • IBM AIX 4.2 or later • HP-UX 10.20 or later • Linux x86 Red Hat 6.0 or later (2.1.3-15 or later of glibc) • SuSE Linux for IBM S/390 and IBM Zseries **For NAI McAfee eBusiness Server 8.5** • Solaris 9 or later **For NAI McAfee eBusiness Server 8.5.1** • Microsoft Windows NT Server version 4.0 or later (Service Pack 6a or later) • Microsoft Windows 2000 Server or Advanced Server (Service Pack 4 or later) • Microsoft Windows Server 2003 • Microsoft Windows XP Professional Version 2002 Service Pack 2 **For Massachusetts Institute of Technology (MIT) Command Line Freeware** • Windows systems: Microsoft Windows NT version 4.0 or later (Service Pack 3 or later), or Microsoft Windows 2000 • UNIX systems: Sun Solaris for SPARC version 2.51 or later IBM AIX 4.2 or later HP-UX 10.20 or later Linux x86 RedHat (RPM) 5.0 or later **For PGP Corporation PGP® Command Line 9.5** • Windows systems: Microsoft Windows XP (SP 2) Microsoft Windows 2003 (SP 1) Microsoft Windows 2000 (SP 4) • UNIX systems: Sun Solaris 9 (SPARC only; x86 is not supported) IBM AIX 5.2 HP-UX 11i Red Hat Enterprise Linux 3.0 on x86 • Mac OS X 10.4 or greater

System name	PGP Package service
Related adapters and services	The PGP Package service works with the following services: • Command Line Adapter 2 • PGP Unpackage service
Application requirements	Before using this service, install one of the following: • McAfee E-Business Server (version 8.1, 8.5, or 8.5.1) from Network Associates Technology, Inc. • PGP Command Line - Freeware (version 6.5.8) previously distributed by MIT (no longer available) • PGP Command Line (version 9.5) from PGP Corporation **Note:** Consider the nature of your PGP usage relative to the PGP vendor's licensing terms when choosing a package.
Initiates business processes?	This service does not initiate business processes. This service cannot be used without a business process.
Invocation	A user who has permission to perform this activity must execute the business process that invokes this service.
Business process context considerations	The configuration parameters and the outgoing documents are picked up by the service in the business process context. In the receiving mode, the service puts the incoming documents into the business process context.
Returned status values	Basic statuses are: • 0 - Success • 1- Error See *Advanced Status Messages* for a list of advanced statuses. Exit Codes will be displayed in the Advanced Status column, pre-pended by [PGPErrorCode].
Restrictions	None
Persistence level	None

System name	PGP Package service

Testing considerations	Create the profile in the PGP Server Manager. This profile stores information about the PGP server, including PGP Type, PGP Executable, PGP Path, the location of the public key ring, the secret key ring, and the random number seed. It enables you to create key maps for secret key sets and conventional key sets.
	
	A pre-defined Command Line Adapter 2 (PGPCmdlineService) is installed with Application. The Command Line Adapter 2 is used for large file support (streaming). Start the remote Command Line 2 client.
	
	To start the remote adapter implementation of the command line adapter:
	1. Locate the client jar (CLA2Client.jar in Install_DIR>/<client>/<cmdline2>) that contains all the necessary classes.
	2. Move the client jar to the machine that has the PGP server installed.
	3. Start the remote adapter implementation using the following command:
	`java -jar CLA2Client.jar <port> [debug]`
	
	For example:
	`java -jar CLA2Client.jar 15699 debug`
	
	Note: The [debug] option is not required.

Implementing the PGP Package Service

To implement the PGP Package service, complete the following tasks:

1. Activate your license for the PGP Package service. See *Managing Services and Adapters*.
2. Create a PGP profile, using the Application PGP Server Manager. See *PGP Server Manager*.
3. Create a PGP Package service configuration. See *Managing Services and Adapters*.
4. Configure the service. See *Configuring the PGP Package Service*.
5. Use the PGP Package service in a business process.

Configuring the PGP Package Service

Before configuring, consider the following:

- public_user (if using Public Key Cryptography) or conv_keymap_name (if using Conventional Cryptography) must be present for PGP Package service to perform encryption.
- secret_keymap_name must be present for PGP Package service to perform signing.
- To perform encryption and signing, a combination of both the previous statements applies.

- If public_user and conv_keymap_name appear in the same business process, public key encryption will take precedence.

To configure the PGP Package service, specify settings specify the settings for the fields in the GPM. These fields are described in the following table:

Field	Description
Config	Name of the service configuration.
workingDir	The working directory where files used for encryption and signing will be read from or written to. Optional if the cmdline2svcname field is defined in the Command Line Adapter 2.
remoteName	Remote name or IP address where the remote adapter implementation is running. Optional if the cmdline2svcname field is defined in the Command Line Adapter 2.
remotePort	Remote port that the remote adapter implementation is listening on. Optional if the cmdline2svcname field is defined in the Command Line Adapter 2.
profile_name	Name of PGP profile from the PGP Server Manager. Required.
compress	Compression to be done before encryption or signing. Valid value is On. Default is On. Required for encryption and signing.
public_user	User name or key ID in the public key ring. Required for encryption (public key cryptography).
secret_keymap_name	Key name defined in the secret key ring in the PGP profile. Required for signing (public key cryptography).
conv_keymap_name	Key name defined in the public key ring in the PGP profile. Required for encryption (conventional cryptography).
conv_cipher	The symmetric cipher to use when performing a conventional encryption operation (that is, conv_keymap_name is used). Valid values are: IDEA, CAST5, 3DES, AES128, AES196, AES256, Twofish. Default is IDEA. Optional.
DocumentId	The document identifier referenced to the document to be processed specifically. The default document for processing is the primary document. Optional.
cmdline2svcname	If not using the default configuration of the Command Line 2 adapter (PGPCmdlineService), enter the name of the configuration to be used. Optional.
ascii_armor	Whether to encode the file with McAfee E-Business Server's base-64 encoding (ASCII-armored format). Valid values are On and Off. Default is On. Optional.

Field	Description
textmode	Whether the input data is ASCII text and should be converted to canonical new lines before encryption. Valid values are On and Off. Default is Off. Optional.
outputfilename	Output file name. For McAfee E-Business Server and PGP Command Line Freeware, outputfilename must have an extension of .asc or .pgp. If a different extension is used, outputfilename will be appended with .asc. For all versions, if outputfilename is not specified, the file name is retrieved from the name of the primary document or the body name of the document and is appended with the following: • *.asc during normal encryption • .exe during sda process • .pga during pgparchive process Optional.
pgp_partner_name	The partner name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected partner profile. Required if you specify a value in the pgp_sponsor_name parameter. The values you specify in the GPM override the values you specify in the profile.
pgp_sponsor_name	The sponsor name used in encryption and signing. If specified, the business process uses the parameters you specify in the selected sponsor profile. Required if you specify a value in the pgp_partner_name parameter. The values you specify in the GPM override the values you specify in the profile.
tmpDir	The directory location for temporary scratch files. If not specified, the temporary files are written in the current working directory. If the shell environmental variable TMP is defined, PGP stores temporary files in the named directory. Optional.

Field	Description
clearsig	Generates a signed message that can be read without PGP. The recipient must still use PGP to verify the signature. Unencrypted PGP-signed messages have a signature certificate pre-pended in binary form. The signed message is compressed. Therefore, it is unreadable by humans even though it is not encrypted. Cannot be used with EncryptAndSign on the command line. If you enable clearsig, it is recommended you enable ascii_armor and textmode also. Valid values are On and Off. Default is Off. Optional.
info	How much information is returned. Valid values are: • Quiet - Only displays error messages. Not applicable to PGP Command Line. If selected defaults to normal mode. • Normal - Displays warnings and error messages. Default. • Verbose - Displays helpful messages, warnings, and error messages. Use this setting to diagnose problems. Only available for McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). If selected with other versions, defaults to normal mode. • Debug - Displays developer-level output in addition to the output produced by the other levels. This level may include the display of internal data, statistics, trace information, and return codes from internal functions. Do not use unless instructed to do so. Not applicable to PGP Command Line. If selected, defaults to normal mode. Optional.

Field	Description
sda	Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified. Creates a self-decrypting executable file, which is conventionally encrypted using a passphrase. The resulting file can be decrypted by double-clicking it and entering the passphrase. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed. SDA files can be created with any platform that McAfee E-Business Server (version 8.1 or later) supports, but can be executed only on Windows platforms. To create sda files with PGP Command Line (version 9.5), set the target_platform parameter (described later in this table). The default file extension is .exe. **Note:** The sda file cannot exceed 4 GB after compression. Valid values are On and Off. Default is Off. Optional.
pgparchive	Applicable only to McAfee E-Business Server (version 8.1 or later) and PGP Command Line (version 9.5). Used only when conv_keymap_name is specified. Creates a file that can be decrypted using the archive reader, which can be redistributed freely. Used to send encrypted files to people who do not have E-Business Server or PGP Command Line installed. The default extension is .pga. Valid values are On and Off. Default is Off. Optional.
discard_paths	Applicable only with sda or pgparchive. Strips relative path information form the list of files in a sda or pgparchive. During the decryption of the archive, the files are placed in the current directory instead of in subdirectories of the current directory. Optional.

Field	Description
target_platform	Applicable only with PGP Command Line (version 9.5) and sda. Specifies the platform an sda file can be decrypted on. Valid values are: • win32 • linux • solaris • aix • hpux • osx Default is the current platform. Optional.

Parameters Passed from Service to BP

The following table contains the parameters that are passed from the PGP Package service to the business process:

Parameter	Description
Action (PGP/Action)	Action of this PGP execution. Valid values are: • ENCRYPT • ENCRYPT_SIGN • SIGN Required.
FileName (PGP/FileName)	Name of the file being processed. Required.
inputFileNamePkg (PGP/inputFileNamePkg)	Name of the file contained in the PGP package. Optional.
Document (PGP/Document)	The processed document is placed in Process Data - not as Primary Document. The attribute is the SCIObjectID, which enables a hyperlink for viewing the content of the processed document. Required.
DocumentId (PGP/DocumentId)	Document identifier of the document. Required.
Status (PGP/Status)	Process status. Valid values are Success and Error. Required.
ErrorCode (PGP/ErrorCode)	Value returned from executing PGP commands. Displayed when the Status is Error. Optional.
ErrorDescription (PGP/ ErrorDescription)	This is the error description based on the ErrorCode. Displayed when the Status is Error. Optional.

Business Process Example - Encrypt Operation (Public Key Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area. The profile is based on PGP107. In this example, you use the default Command Line2 adapter configuration, PGPCmdlineService, to execute the encrypt command. You want to use the working directory, remote name and port stated in the BPML. Therefore, these values override the pre-configured values in PGPCmdLineService. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption.

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
              <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="public_user">0x2343</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt Operation (Conventional Encryption)

This following business process uses the PGP Package service to encrypt the primary document in the document area of process data. The profile is based on PGP107. In this example, you use the Command Line2 adapter configuration, MyCLA2, to execute the commands. The remote name, port, and working directory are pre-configured in the service configuration. The value of conv_keymap_name, Conv_abc_tp, which must be in the profile's conventional key map, is used for conventional encryption:

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="conv_keymap_name">Conv_abc_tp</assign>
              <assign to="conv_cipher">CAST5</assign>
              <assign to="cmdline2svcname">MyCLA2</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt and Sign Operation (Public Key Encryption)

The following business process uses the PGP Package service to encrypt and sign the primary document in the document area. For signing, you need to pass in the secret_keymap_name, which must be in the PGP107 profile's secret key map. The public key ID, which must be in the public keyring file specified in the profile, PGP107, is used for encryption. In this example, you choose not to compress the document before signing and encryption.

```
<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
        <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">off</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="public_user">0x2343</assign>
              <assign to="secret_keymap_name">my_secret</assign>
</output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt and Sign Operation (Conventional Encryption)

The following business process uses PGP Package Service to encrypt and sign the Primary Document in the document area. For signing, the user needs to pass in the secret_keymap_name, which must be present in the PGP107 profile's Secret Key Map. The value of conv_keymap_name, Conv_abc_tp, which must be present in the Profile's Conventional Key Map, is used for conventional encryption. The user chooses not to compress the document before signing and encryption.

```
<process name="PGP_Encrypt_Sign">
  <sequence name="optional">
    <operation name="One">
      <participant name=" PGPPackageService "/>
      <output message="Xout">
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">off</assign>
              <assign to="workingDir">/localsvr/share/tmp</assign>
              <assign to="remoteName">nn.nnn.nn.nnn</assign>
              <assign to="remotePort">xxxxx</assign>
              <assign to="conv_keymap_name">Conv_abc_tp</assign>
              <assign to="conv_cipher">CAST5</assign>
              <assign to="secret_keymap_name">si_secret</assign>
              <assign to="." from="*"></assign>
     </output>
     <input message="Xin">
        <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Encrypt Operation (Public Key Encryption) Using a Specific Document ID

The following business process uses the PGP Package service to encrypt a document, with the document ID columbia:1774b9b:feaea8ae12:-6ea8 in the document area.

```
<process name="PGP_Encrypt ">
  <sequence name="optional">
    <operation name="One"> PGPPackageService
      <participant name="PGPPackageService"/>
      <output message="Xout">
              <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="public_user">0x2343</assign>
              <assign to="DocumentId">columbia:1774b9b:feaea8ae12:-6ea8</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - Sign Operation

The following business process uses the PGP Package service to sign the primary document in the document area.

```
<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
        <assign to="." from="*"></assign>
              <assign to="profile_name">PGP107</assign>
              <assign to="compress">on</assign>
              <assign to="workingDir">/server1/tmp</assign>
              <assign to="remoteName">00.000.00.000</assign>
              <assign to="remotePort">12345</assign>
              <assign to="secret_keymap_name">my_secret</assign>
     </output>
     <input message="Xin">
              <assign to="." from="*"></assign>
     </input>
    </operation>
  </sequence>
</process>
```

Business Process Example - OnFault Handling

The following business process shows the onFault handling for the PGP Package service.

```
<process name="PGP_Sign ">
  <sequence name="optional">
    <operation name="One">
      <participant name="PGPPackageService"/>
      <output message="Xout">
   <assign to="profile_name">PGP107</assign>
   <assign to="compress">on</assign>
   <assign to="workingDir">/localsvr/share/tmp</assign>
```

```
   <assign to="remoteName">nn.nnn.nn.nnn</assign>
   <assign to="remotePort">12345</assign>
   <assign to="secret_keymap_name">si_secret</assign>
   <assign to="." from="*"></assign>
     </output>
     <input message="Xin">
        <assign to="." from="*"></assign>
     </input>
    </operation>
    <assign to="Status">The file is signed successfully</assign>
    <onFault>
      <assign to="Status">General Error Occurred</assign>
    </onFault>
    <onFault code="[PGPErrorCode] Signature Check error">
      <assign to="Status">Incorrect signature</assign>
    </onFault>
  </sequence>
</process>
```

Business Process Example - PGP Partner and PGP Sponsor

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

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

Advanced Status Messages

The following table contains exit codes from the McAfee E-Business Server and PGP Command Line Freeware. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	Exit OK, no error
1	Invalid file
2	File not found
3	Unknown file
4	Batch mode error
5	Bad argument
6	Process Interrupted
7	Out of memory error
8	Environment error
20	Signature error
21	Public Key Encryption error
22	Encryption error

Status	Description
23	Compression error
30	Signature Check error
31	Public Key Decryption error
32	Decryption error
33	Decompression error
34	Keyring locked error
101	File parsing error

The following table contains exit codes from PGP Command Line (version 9.5) from PGP Corporation. The content of the Description field is displayed in the Advanced Status column, preceded by [PGPErrorCode]:

Status	Description
0	PGP Command Line exited successfully.
64	Parser error.
71	Bad data was received from the operating system at startup.
128	An internal error occurred.
129	An initialization failure occurred on startup.
130	A user interrupt occurred.
145	Error purging a cache: passphrase, keyring, or both.
146	Error creating keyring files.
147	Error during a speed test operation.
160	Complete failure during a file wipe.
161	Partial fail, partial success during a file wipe (one file wiped, one not, for example).
162	Complete failure during an encode.
163	Partial failure during an encode.
164	Complete failure during a decode.
165	Partial failure during a decode.
210	Error during one of the key list operations.
220	Error during key maintenance.
221	Error when checking signatures.
222	Error when checking user IDs.
230	Error during one of the key edit operations.
240	Error during one of the key server operations.
245	Error with supplied license.
251	License is expired.
255	An unknown error occurred.

The following table contains errors that result from the PGP Package service when it validates information before executing PGP commands on the remote server. The content of the status field will be displayed in the Advanced Status column:

Status	Description
Error in accessing the document with a given DocumentId.	The DocumentId value given in the BPML is incorrect.
Fail to get data from Primary Document. There is no Primary Document.	Primary Document is mandatory.
Incorrect Profile Name in BPML Param: 'profile_name'. It is not found in the PGP Server Manager.	The profile_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'secret_keymap_name'). It is not found in the PGP Profile's Secret KeyMap.	The secret_keymap_name value given in the BPML is incorrect.
Incorrect Key Name (BPML Param: 'conv_keymap_name'). It is not found in the PGP Profile's Conventional KeyMap.	The conv_keymap_name value given in the BPML is incorrect.

Translation Service

The following table provides an overview of the Translation service:

System name	Translation Type
Graphical Process Modeler (GPM) categories	All Services, Translation
Description	Performs translation of the primary document using a specified map, and replaces the primary document with the result of the translation.
Business usage	Performs translation of the primary document within business processes.
Usage example	You want to take positional data from your order system and translate it to variable-length-delimited data so that it can be read by your billing system. Use the application Map Editor to create a map that will translate the incoming data from positional data to variable-length-delimited data. Write a business process that will put the data into the primary document, then start the Translation service. Using the map you created, the service translates the data from positional data to variable-length-delimited, and replaces the old data with newly translated data in the primary document.
Preconfigured?	There is a configuration of the service delivered with the application, but you must configure parameters for it in the GPM.
Requires third party files?	No
Platform availability	All supported application platforms.
Related services	No

System name	Translation Type
Application requirements	The map specified in the map_name parameter must be registered with the application and activated. If either of these conditions is not met then the translation will not be performed.
Initiates business processes?	No
Invocation	Runs as part of a business process.
Business process context considerations	The Translation service looks for the following parameters in the business process context. If the service finds them, it uses them during translations where either the input or output is EDI: • edi_output_tag_delimeter • edi_output_segment_delimeter • edi_output_element_delimeter • edi_output_sub_element_delimeter • edi_output_repeating_element_delimeter • edi_output_release_character • edi_output_decimal_separator • edi_input_tag_delimeter • edi_input_segment_delimeter • edi_input_element_delimeter • edi_input_sub_element_delimeter • edi_input_repeating_element_delimeter • edi_input_release_character • edi_input_decimal_separator
Returned status values	• Success - Translation was successful. • Error - Errors were encountered during translation or translation could not be performed. See the Translator report contained in the Business Process Context Status report for further detail.
Restrictions	No
Persistence level	None
Testing considerations	The best way to test is within a simple business process where the Translation service is the only operation. After the business process runs, verify the output in the application, and review the translator report for detail on what occurred during the translation.

How the Translation Service Works

The Translation service translates data in the following file formats:

- Electronic data interchange (EDI)
- Positional
- Variable-length-delimited
- Extensible Markup Language (XML)

- Structured Query Language (SQL)
- Japanese Center for Informatization of Industry (CII)

Note: If the input document character encoding is specified in the application, it overrides the encoding specified in the map. The output document content type and character encoding are set based on the information contained in the map.

The Translation service creates a translation report.

Implementing the Translation Service

To implement the Translation service, complete the following tasks:

1. Activate your license for the Translation service. See *An Overview of Implementing Services*.
2. If you are using a map that has a database on the output side, you must set up a connection to the database that contains the tables you want to access. See *Setting Up a Connection to an External Database*.
3. Create a Translation service configuration. See *Creating a Service Configuration*.
4. Configure the Translation service. See *Configuring the Translation Service*.
5. Use the Translation service in a business process.

Configuring the Translation Service

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

Field	Description
Config	Name of the service configuration.
edi_input_decimal_separator	Character used to indicate the decimal point on the input side.
edi_input_element_delimiter	Character used to delimit elements (fields) on the input side.
edi_input_release_character	Character used to quote elements (fields) that contain the delimiter on the input side.
edi_input_repeating_element_delimiter	Character used to delimit repeating elements on the input side.
edi_input_segment_delimiter	Character used to delimit segments on the input side.
edi_input_sub_element_delimiter	Character used to delimit sub-elements on the input side.
edi_input_tag_delimiter	Character used to delimit tags on the input side.
edi_output_decimal_separator	Character used to indicate the decimal point on the output side.
edi_output_element_delimiter	Character used to delimit elements (fields) on the output side.
edi_output_release_character	Character used to quote elements (fields) that contain the delimiter on the output side.
edi_output_repeating_element_delimiter	Character used to delimit repeating elements on the output side.

Field	Description
edi_output_segment_delimiter	Character used to delimit segments on the output side.
edi_output_sub_element_delimiter	Character used to delimit sub-elements on the output side.
edi_output_tag_delimiter	Character used to delimit tags on the output side.
exhaust_input	Whether to execute the map until the Translation service has translated all of the input. Valid values are Yes and No. **Note:** If your map design is faulty (that is, if the data structure does not match the layout of the map), the data in the input file cannot be properly processed. If a segment is present in the input file it must be defined and active in the map and in the proper sequence. When the translator reads a segment, it tries to match it to the records in the map based on their tag values. If exhaust_input is set to "Yes" the translator attempts to match each segment in the input file to a segment in the map, until it reaches the end of the input file. Conversely, if exhaust_input is set to "No," the translator does not re-invoke the map to continue processing the remaining data in the input file.
map_name	Name of the map used for translation. The map must already be checked in to the application and enabled.
output_report_to_process_data	Whether to output the report to process data. Valid values are: • Yes: Output the report to process data. • No: Do no output the report to process data.
output_to_process_data	Whether the output of the translation should be placed in the process data tree. The output must be XML. Valid values are Yes and No.
useStreams	Whether to support large files (streaming mode). Valid values are Yes (default), No, and blank (which uses default). The default was changed with release 4.1.1, patch 1973. In versions previous to that, the service did not use document streaming by default.
validate_input	Validates the input to the input side of the map. Valid values are Yes and No.
validate_input_against_dtd	Validates the input to the DTD specified in the input document. Valid values are No validation, Validate using a DTD, and Validate using an XML schema.

Field	Description
validate_output	Validates the output to the output side of the map. Valid values are Yes and No.

Parameters Passed Through BPML Only

The following parameters can be passed through BPML using an Assign statement. Note that these parameters are not available through the GPM.

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. **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: • 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: FromSchema="from1@from2" ToSchema="to1@to2" In this example, any schema names matching "from1" are changed to "to1," and any schema names matching "from2" are changed to "to2." 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: FromSchema="from1@from2@from3" ToSchema="to" In this example, any schema names matching "from1," "from2," or "from3" will be changed to "to." • 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. • 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.

Parameter	Description
sql_statement_use_batching	Whether to enable batching. Valid values are Yes and No (default).
sql_statement_maximum_ batchsize	The maximum size of the SQL statement batch. Only valid if sql_statement_use_batching is set to Yes.

Turning on SQL Statement Batching

In your map, any record for which the On failure, automatically switch selected operation and retry Inserts as Updates or Updates as Inserts setting is turned on (enabled) is not be batched, because batching is not supported for records that have retry enabled. For these records, the SQL is executed with no batching, and records that do not have retry enabled are batched.

Additionally, in the map, the data source must have **Use Transaction** enabled. If **Use Transaction** is turned off, then batching is not performed.

Finally, the database must support batching. If the database does not support batching, the batch service parameters will be ignored and the SQL statements will not be batched.

This example BPML demonstrates how you might enable SQL statement batching:

```
<operation name="Translation">
      <participant name="Translation"/>
      <output message="TranslationTypeInputMessage">
        <assign to="map_name">insert</assign>
        <assign to="sql_statement_use_batching" from="'yes'"/>
        <assign to="sql_statement_maximum_batchsize" from="'500'"/>
        <assign to="." from="*"></assign>
      </output>
      <input message="in">
        <assign to="." from="*"></assign>
      </input>
    </operation>
```

XML Digital Signature Service

The following table provides an overview of the XML Digital Signature service:

Service Name	XML Digital Signature Service
System name	XMLDSigService
Graphical Process Modeler (GPM) Category	All Services
Description	Use the XML Digital Signature service to compose and verify digital signatures.
Business usage	Use this service to create enveloped, enveloping, detached, and a combination of all three signatures.
Usage example	A business process that needs a document to be digitally signed or verified can invoke this service by passing the required parameters.
Preconfigured?	No

Service Name	XML Digital Signature Service
Requires third party files?	Yes. Requires xss4j.jar. This is preloaded in the Application.
Platform availability	All supported Application platforms.
Related services	N/A
Application requirements	N/A
Initiates business processes?	No
Invocation	This service is invoked from a business process.
Business process context considerations	You must be familiar with the Internal Service (WF/BP parameters) that invokes this adapter. The WF parameters are the values passed into the Internal Service and BP parameters are the values specified within the business process code.
Returned status values	• signRequest • verifyRequest
Restrictions	None
Persistence level	System Default
Testing considerations	You should use the correct certificates for signing. The most common problem encountered is that certificates used for signing are not created with a storepass value and a keypass value of integrator. If you receive an error with this condition, see your system administrator.
Restrictions	None

How the XML Digital Signature Service Works

The XML Digital Signature service signs or verifies the XML signature. It provides integrity and confidentiality of XML documents and messages.

The XML Digital Signature service in the Application supports the following types of XML signatures:

- Enveloped (default) - signature of either an entire document or a document fragment where the XML signature is embedded within the signed document.
- Enveloping - signature where signed data is embedded within XML signature structure.
- Detached - signature where the signed entities are not attached to the actual signature fragment.

  Note: In the Application, a detached signature type signs on the detached workflow document. The Reference URI of the detached document is the document ID.
- Combination (combination of enveloped, enveloping, and detached)

Implementing the XML Digital Signature Service

To implement the XML Digital Signature service, complete the following tasks:

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

System Administrator Tasks

The following procedure describes the system administrator tasks for XML Digital Signature service.

Importing a KeyCert into Application

1. Login to Application.
2. Select **Trading Partner** -> **Digital Certificates** -> **Trusted**.
3. Select **New Certificate** under Check in.
4. Select the certificate and click **Next**.
5. Enter the Certificate Name and click **Next**.
6. Review and click **Finish**.
7. You can use this certificate in your BPML associated with the appropriate field (signCertificateIdentifier).

Configuring the XML Digital Signature Service

To configure the XML Digital Signature service, you must specify settings for the following fields:

Field	Description
Name	XML Digital Signature Service
Description	Signs and validates XML digital signatures
Select a Group	Select one of the following options: • None - You do not want to include this configuration in a group at this time. • Create New Group - You can enter a name for a new group in this field, which will then be created along with this configuration. • Select Group - If you have already created one or more groups for this service type, they are displayed in the list. Select a group from the list. **Note:** For more information about groups, see *Managing Services and Adapters*.

Output from Business Process to Service

The following table contains the parameters passed from the business process to the XML Digital Signature service when it invokes with the output message set to `signRequest`:

Parameter	Description
action	Required. The required action. The value can be a valid string. Valid value - sign.
signatureType	Required. The type of signature. Valid values are: • enveloped (default) • enveloping • detached • combination of enveloped, enveloping, and detached
signCertificateIdentifier	Required. The alias of a private key in the certificate.
certificateIdentifier	Optional. The alias of a public key in the certificate. When this parameter is used during signing, the KeyInfo element must be included in the signature.
nodeToSign	Optional. Indicates the node that needs to be signed. This parameter is used when signing XML document and the node exists in the document to be signed. If this parameter is not specified, the whole document is signed. Multiple nodes can be specified using comma (,) as delimiter. When signing with enveloped signature, the node to be signed should contain `ID` attribute. Valid node names - node1, node2

Transforms	Optional. The required Transforms to be used when signing. If omitted when signing with enveloped signature type, the enveloped-signature Transform will be used.
	
	If omitted when signing non-XML document with enveloping signature type, the base64 Transform will be used.
	
	An error will be thrown when the transform algorithm is invalid in xss4j.
	
	A valid example using Transform:
	`<Transforms>`
	`<Transform Algorithm="...">`
	`</Transform>`
	`</Transforms>`

Parameter	Description

documents	Optional. Used to sign multiple documents or when primary document is empty. This node contains a list of document nodes.
	
	This parameter is required when primary document is empty.
	
	Each document node contains:
	• documentID - Required for all signature types.
	• nodeToSign - Required when signing XML document and optional for all signature types.
	• signatureType - Required for combination signature.
	• Transforms/Transform - Optional for all signature types and can contain multiple `Transform` nodes with different algorithms.
	
	A valid example:
	`<documents>`
	`<document>`
	`<documentID>`
	`xxx`
	`</documentID> <signatureType>`
	`xxx`
	`</signatureType>`
	`<nodeToSign>`
	`node1, node2`
	`</nodeToSign>`
	`<Transforms>`
	`<Transform`
	`Algorithm="...">`
	`</Transform>`
	`</Transforms>`
	`</document>`
	`</documents>`

The following table contains the parameters passed from the business process to the XML Digital Signature service when it invokes with the output message set to `verifyRequest`:

Parameter	Description
action	Required. The required action. The value can be a valid string. Valid value - verify.
certificateIdentifier	Optional. The alias of a public key in the certificate. If certificateIdentifier is not present, the certificate information is retrieved from the KeyInfo element of signature.

Parameter	Description

documents	Optional. Used for detached signature verification. It contains a list of document nodes and each document contains one documentID. The sequence of the detached document list should follow the reference sequence in XML signature. This parameter is not applicable for verifying enveloped and enveloping signature.
	
	A valid example:
	
	`<documents>`
	`<document>`
	`<documentID>`
	`xxx`
	`</documentID>`
	`</document>`
	`</documents>`

Business Process Examples

The following example business processes illustrate using the XML Digital Signature service:

Example Business Process 1

The following BPML signs the document based on the parameters passed from BPML to the XML Digital Signature service.

```
<process name="xmldsig_enveloped">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
            <assign to="." from="*"/>
            <assign to="action">sign</assign>
            <assign to="signatureType">enveloped</assign>
            <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      </output>
      <input message="signResponse">
            <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
            <assign to="." from="*"/>
            <assign to="action">verify</assign>
            <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
            <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

Example Business Process 2

The following BPML shows how to sign the Primary Document and add the KeyInfo element within the Signature element by including the certificateIdentifier parameter in the signing request. The example also includes how to construct the Transforms node.

```
<process name="xmldsig_enveloped_transform_keyinfo">
  <sequence>
    <assign
to="temp/@Algorithm">http://www.w3.org/2000/09/xmldsig#enveloped-signature</assign>
     <assign to="Transforms/Transform" from="temp/@*"/>

     <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
           <assign to="." from="*"/>
           <assign to="action">sign</assign>
           <assign to="signatureType">enveloped</assign>
           <assign to="signCertificateIdentifier">test_rsa_priv</assign>
           <assign to="certificateIdentifier">test_rsa_pub</assign>
           <assign to="Transforms" from="Transforms/node()"/>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
           <assign to="." from="*"/>
           <assign to="action">verify</assign>
      </output>
      <input message="verifyResponse">
           <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

Example Business Process 3

The following BPML shows how to sign particular nodes in the Primary Document. The nodes to be signed are delimited by comma (,).The signature type is enveloped.

```
<process name="xmldsig_enveloped_nodetosign">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
           <assign to="." from="*"/>
           <assign to="action">sign</assign>
           <assign to="nodeToSign">value1,value2</assign>
           <assign to="signatureType">enveloped</assign>
           <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      </output>
      <input message="signResponse">
           <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
```

```
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
           <assign to="." from="*"/>
           <assign to="action">verify</assign>
           <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

Example Business Process 4

The following BPML shows how to sign particular nodes in the Primary Document. The nodes to be signed are delimited by comma (,). The signature type is enveloping.

```
<process name="xmldsig_enveloping_nodetosign">
  <sequence>

    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
          <assign to="." from="*"/>
          <assign to="action">sign</assign>
          <assign to="nodeToSign">value1,value2</assign>
          <assign to="signatureType">enveloping</assign>
          <assign to="signCertificateIdentifier">test_rsa_priv</assign>
      </output>
      <input message="signResponse">
          <assign to="." from="*"></assign>
      </input>
    </operation>

    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
          <assign to="." from="*"/>
          <assign to="action">verify</assign>
          <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>

  </sequence>
</process>
```

Example Business Process 5

This following input file and BPML shows how to sign multiple documents passed from "documents/document" parameter. The nodes to be signed are delimited by comma (,). The signature type is enveloping.

Input file:

```
<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <nodeToSign>value1,value2</nodeToSign>
    <Transforms>
      <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
```

```
        <XPath>descendant-or-self::Contract</XPath>
      </Transform>
    </Transforms>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
  </document>
</documents>
```

BPML:

```
<process name="xmldsig_enveloping_documents">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">documents</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET">PrimaryDocument</assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>
    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*"></assign>
        <assign to="action">sign</assign>
        <assign to="signatureType">enveloping</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="documents" from="documents/node()"></assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET" from="'documents'"></assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>
    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
        <assign to="." from="*"></assign>
        <assign to="action">verify</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
   </sequence>
</process>
```

Example Business Process 6

The following input file and BPML shows how to sign multiple documents passed from "documents/document" parameter, Transforms/Transform/XPath is used to sign specific node. The signature type is detached.

Input file:

```
<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <Transforms>
      <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
         <XPath>descendant-or-self::Contract</XPath>
      </Transform>
    </Transforms>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
  </document>
</documents>
```

BPML:

```
<process name="xmldsig_enveloping_documents">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">documents</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET">PrimaryDocument</assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>
    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*"></assign>
        <assign to="action">sign</assign>
        <assign to="signatureType">detached</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="documents" from="documents/node()"></assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET" from="'documents'"></assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>
```

```
    <assign to="document/documentID">sgconrad:-628e3b67:11569be511e:-6d7a</assign>
    <assign to="documents/document" from="document/node()" append="true"></assign>
    <assign to="document/documentID">sgconrad:-628e3b67:11569be511e:-682f</assign>
    <assign to="documents/document" from="document/node()" append="true"></assign>
    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
        <assign to="." from="*"></assign>
        <assign to="action">verify</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
   </sequence>
</process>
```

Example Business Process 7

The following input file and BPML shows how to sign multiple documents passed from "documents/document" parameter. The signature type is combination.

Input file:

```
<documents>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75fc</documentID>
    <signatureType>enveloped</signatureType>
    <nodeToSign>value1,value2</nodeToSign>
  </document>
  <document>
    <documentID>sgconrad:-3c3ab664:1158cfc1d5d:-5431</documentID>
    <signatureType>enveloping</signatureType>
    <Transforms>
      <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#base64">
      </Transform>
    </Transforms>
  </document>
  <document>
    <documentID>sgconrad:31e5343c:1158d3b080f:-75bc</documentID>
    <signatureType>detached</signatureType>
    <nodeToSign>node1,node2</nodeToSign>
  </document>
</documents>
```

BPML:

```
<process name="xmldsig_enveloping_documents">
  <sequence>
    <operation name="XML Encoder">
      <participant name="XMLEncoder"/>
      <output message="XMLEncoderTypeInputMessage">
        <assign to="output_to_process_data">YES</assign>
        <assign to="mode">xml_to_process_data</assign>
        <assign to="root_element">documents</assign>
        <assign to="." from="*"></assign>
      </output>
      <input message="inmsg">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET">PrimaryDocument</assign>
      </output>
```

```
      <input message="releaseResponse">
      </input>
    </operation>
    <operation name="SignMessage">
      <participant name="XMLDSigService"/>
      <output message="signRequest">
        <assign to="." from="*"></assign>
        <assign to="action">sign</assign>
        <assign to="signatureType">combination</assign>
        <assign to="signCertificateIdentifier">test_rsa_priv</assign>
        <assign to="documents" from="documents/node()"></assign>
      </output>
      <input message="signResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
    <operation>
      <participant name="ReleaseService"/>
      <output message="releaseRequest">
        <assign to="TARGET" from="'documents'"></assign>
      </output>
      <input message="releaseResponse">
      </input>
    </operation>
    <assign to="document/documentID">sgconrad:-628e3b67:11569be511e:-6d7a</assign>
    <assign to="documents/document" from="document/node()" append="true"></assign>
    <operation name="VerifyMessage">
      <participant name="XMLDSigService"/>
      <output message="verifyRequest">
        <assign to="." from="*"></assign>
        <assign to="action">verify</assign>
        <assign to="certificateIdentifier">test_rsa_pub</assign>
      </output>
      <input message="verifyResponse">
        <assign to="." from="*"></assign>
      </input>
    </operation>
   </sequence>
</process>
```

XML Digital Signature Service Examples

The following example signature types illustrate using the XML Digital Signature service:

Example of Enveloped Signature

The following example shows the enveloped signature type:

```
<?xml version="1.0" encoding="UTF-8"?>
<test>
 <value1 ID="1">
testval1</value1>
 <value2 ID="2">
testval2</value2>
 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
   <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
   <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
   <Reference URI="">
    <Transforms>
     <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
```

```
fgWg+5VNk9ZqUy86IOIE3QS+bGQ=</DigestValue>
   </Reference>
  </SignedInfo>
  <SignatureValue>
MYWSLB0T9qX+Qu2BnMWpGC4KmKT5pws0M9GtQQsMGYT5rxXCMe2qeMKAV65WoY0ib7mTxjj9
nChQSdyN1yYYn6nRvvKl9lX8LWVNpJj58lE7/alohe8/+0hOAWKxBsiFDhBbUC0jHG1+Qp2f
M3KAe2kyBUPjN57isAmmH64y/LI= </SignatureValue>
  <KeyInfo>
   <KeyValue>
    <RSAKeyValue>
     <Modulus>
kFhwg4m9hjFmrlxVR3w0XmYx7fgGsoh+aelmXlzCug5gRV0t0XeSpaeoX1jXu6gacJ
Vl/pOlNs+av+iviDKmS94LDPJtjAcl7C9dZbbt39N+/2S9WBAtJGXk5MOIu0aab50D
UFK55mUbpsZzwoVQrisW+KArnWlbrUP5xWXsnwM= </Modulus>
     <Exponent>
AQAB</Exponent>
    </RSAKeyValue>
   </KeyValue>
   <X509Data>
    <X509IssuerSerial>
     <X509IssuerName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509IssuerName
>
     <X509SerialNumber>
1190704157</X509SerialNumber>
    </X509IssuerSerial>
    <X509SubjectName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509SubjectNam
e>
    <X509Certificate>
MIICBjCCAW8CBEb4tB0wDQYJKoZIhvcNAQEFBQAwSTElMCMGCSqGSIb3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvbTELMAkGA1UEBhMCU0cxEzARBgNVBAMTCnNlcmVuYV9yc2EwIBcNMDcwOTI1
MDcwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEkxJTAjBgkqhkiG9w0BCQEWFnNlcmVuYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTAlNHMRMwEQYDVQQDEwpzZXJlbmFfcnNhMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5Klp6hf
WNe7qBpwlWX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL11ltu3f037/ZL1YEC0kZeTkw4i7RppvnQNQ
UrnmZRumxnPChVCuKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAC+7glCs
TKBSURkwmbA4k/SYVOOhhz3VkBX0he3r1/Vd6Qk8IlRJwQj5AT8e40gz+vqO0GvjaYAx70bvIGqn
yYE/VVJJOG5Zw6Tott69Dx4AOCrmbZb96zOAjlcEI3Ol7U1h+9+Uo2h5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
   </X509Data>
  </KeyInfo>
 </Signature>
</test>
```

Example of Enveloped Signature with particular nodes

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

```
<?xml version="1.0" encoding="UTF-8"?>
<test>
 <value1 ID="1">
testval1</value1>
 <value2 ID="2">
testval2</value2>
 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
   <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
   <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
   <Reference URI="#1">
    <Transforms>
     <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
XEZm5p6sjqi+Gt61BHQTOWtL2Nw=</DigestValue>
```

```
   </Reference>
   <Reference URI="#2">
    <Transforms>
     <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
1k1M9eAUMUWM4FP4OA1/YC9qS/Y=</DigestValue>
   </Reference>
  </SignedInfo>
  <SignatureValue>
UmZlQ5g4ummBZu4IRg+BHDpZeeX4ix1GXODhyG+f5tvw+4ZDKIGn8qm5wdAj3+/wJS84Lx/V
hRgjyaucMjM3qEoMAEXEWYRTag0PdeOmfrSUffictYcTaGpPUYjDiWfKyFXAlCL5XmSeBW5s
MGgy5Bi/Y0HB1gwlmu+Jsh6FsGM= </SignatureValue>
  <KeyInfo>
   <KeyValue>
    <RSAKeyValue>
     <Modulus>
kFhwg4m9hjFmrlxVR3w0XmYx7fgGsoh+aelmXlzCug5gRV0t0XeSpaeoX1jXu6gacJ
Vl/pOlNs+av+iviDKmS94LDPJtjAcl7C9dZbbt39N+/2S9WBAtJGXk5MOIu0aab50D
UFK55mUbpsZzwoVQrisW+KArnWlbrUP5xWXsnwM= </Modulus>
     <Exponent>
AQAB</Exponent>
    </RSAKeyValue>
   </KeyValue>
   <X509Data>
    <X509IssuerSerial>
     <X509IssuerName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509IssuerName
>
     <X509SerialNumber>
1190704157</X509SerialNumber>
    </X509IssuerSerial>
    <X509SubjectName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509SubjectNam
e>
    <X509Certificate>
MIICBjCCAW8CBEb4tB0wDQYJKoZIhvcNAQEFBQAwSTElMCMGCSqGSIb3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvbTELMAkGA1UEBhMCU0cxEzARBgNVBAMTCnNlcmVuYV9yc2EwIBcNMDcwOTI1
MDcwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEkxJTAjBgkqhkiG9w0BCQEWFnNlcmVuYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTAlNHMRMwEQYDVQQDEwpzZXJlbmFfcnNhMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5Klp6hf
WNe7qBpwlWX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL11ltu3f037/ZL1YEC0kZeTkw4i7RppvnQNQ
UrnmZRumxnPChVCuKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAC+7glCs
TKBSURkwmbA4k/SYVOOhhz3VkBX0he3r1/Vd6Qk8IlRJwQj5AT8e40gz+vqO0GvjaYAx70bvIGqn
yYE/VVJJOG5Zw6Tott69Dx4AOCrmbZb96zOAjlcEI3Ol7U1h+9+Uo2h5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
   </X509Data>
  </KeyInfo>
 </Signature>
</test>
```

Example of Enveloping Signature

The following example shows the enveloping signature type:

```
<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
  <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <Reference URI="#test">
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
   <DigestValue>
ILdA0tPr6GHE9C2/aLX4GQZc3XI=</DigestValue>
  </Reference>
```

```
 </SignedInfo>
 <SignatureValue>
gJ5H1D5gYydhG5NgFitWeiDs+K49CoFyauodfGG8m1vnBfCfPieu14dt4CG2/w7OxbsS5Kjy
J8+iHePiaVxdu3xYJu0ox4UkCK/wwxvnXzWv+AlS+Kj2fwkvzp6++auDOSC2Ucwe1ncwZFX5
xCSNq9wQYH12aULKsVB/bYtcvwo= </SignatureValue>
 <KeyInfo>
  <KeyValue>
   <RSAKeyValue>
    <Modulus>
kFhwg4m9hjFmrlxVR3w0XmYx7fgGsoh+aelmXlzCug5gRV0t0XeSpaeoX1jXu6gacJ
Vl/pOlNs+av+iviDKmS94LDPJtjAcl7C9dZbbt39N+/2S9WBAtJGXk5MOIu0aab50D
UFK55mUbpsZzwoVQrisW+KArnWlbrUP5xWXsnwM= </Modulus>
    <Exponent>
AQAB</Exponent>
   </RSAKeyValue>
  </KeyValue>
  <X509Data>
   <X509IssuerSerial>
    <X509IssuerName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509IssuerName
>
    <X509SerialNumber>
1190704157</X509SerialNumber>
   </X509IssuerSerial>
   <X509SubjectName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509SubjectNam
e>
   <X509Certificate>
MIICBjCCAW8CBEb4tB0wDQYJKoZIhvcNAQEFBQAwSTElMCMGCSqGSIb3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvbTELMAkGA1UEBhMCU0cxEzARBgNVBAMTCnNlcmVuYV9yc2EwIBcNMDcwOTI1
MDcwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEkxJTAjBgkqhkiG9w0BCQEWFnNlcmVuYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTAlNHMRMwEQYDVQQDEwpzZXJlbmFfcnNhMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5Klp6hf
WNe7qBpwlWX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL11ltu3f037/ZL1YEC0kZeTkw4i7RppvnQNQ
UrnmZRumxnPChVCuKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAC+7glCs
TKBSURkwmbA4k/SYVOOhhz3VkBX0he3r1/Vd6Qk8IlRJwQj5AT8e40gz+vqO0GvjaYAx70bvIGqn
yYE/VVJJOG5Zw6Tott69Dx4AOCrmbZb96zOAjlcEI3Ol7U1h+9+Uo2h5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
  </X509Data>
 </KeyInfo>
 <Object xmlns="" Id="test">
  <test>
   <value1 ID="1">
testval1</value1>
   <value2 ID="2">
testval2</value2>
  </test>
 </Object>
</Signature>
```

Example of Enveloping Signature with particular nodes

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

```
<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
  <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <Reference URI="#value1">
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
   <DigestValue>
mb5wQZvkO1C4+YzJUQ0Q2eL1nNg=</DigestValue>
  </Reference>
  <Reference URI="#value2">
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
```

```
   <DigestValue>
cITL2XfG2q9roG/XpFaoa/JSiGU=</DigestValue>
  </Reference>
 </SignedInfo>
 <SignatureValue>
Z6hxQRWv1+RZMU5UYIN06LmhFfGVtNIniIfeAmNy5TGX6SZAY5vgYbuhhZtq+LWG5nwzSQLX
Kv7TIb8N++LYgJeUtcwumCvI6t6rl6PJuSQiJZnucdpGxChrE1rra2WrRZYXxaLkoSUrBJc1
pCnXscU6F0eHmKakIV0ZypdFZT4= </SignatureValue>
 <KeyInfo>
  <KeyValue>
   <RSAKeyValue>
    <Modulus>
kFhwg4m9hjFmrlxVR3w0XmYx7fgGsoh+aelmXlzCug5gRV0t0XeSpaeoX1jXu6gacJ
Vl/pOlNs+av+iviDKmS94LDPJtjAcl7C9dZbbt39N+/2S9WBAtJGXk5MOIu0aab50D
UFK55mUbpsZzwoVQrisW+KArnWlbrUP5xWXsnwM= </Modulus>
    <Exponent>
AQAB</Exponent>
   </RSAKeyValue>
  </KeyValue>
  <X509Data>
   <X509IssuerSerial>
    <X509IssuerName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509IssuerName
>
    <X509SerialNumber>
1190704157</X509SerialNumber>
   </X509IssuerSerial>
   <X509SubjectName>
CommonName=serena_rsa,Country=SG,EmailAddress=serena_li@stercomm.com</X509SubjectNam
e>
   <X509Certificate>
MIICBjCCAW8CBEb4tB0wDQYJKoZIhvcNAQEFBQAwSTElMCMGCSqGSIb3DQEJARYWc2VyZW5hX2xp
QHN0ZXJjb21tLmNvbTELMAkGA1UEBhMCU0cxEzARBgNVBAMTCnNlcmVuYV9yc2EwIBcNMDcwOTI1
MDcwOTE3WhgPMjA2MjA2MjgwNzA5MTdaMEkxJTAjBgkqhkiG9w0BCQEWFnNlcmVuYV9saUBzdGVy
Y29tbS5jb20xCzAJBgNVBAYTAlNHMRMwEQYDVQQDEwpzZXJlbmFfcnNhMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCQWHCDib2GMWauXFVHfDReZjHt+AayiH5p6WZeXMK6DmBFXS3Rd5Klp6hf
WNe7qBpwlWX+k6U2z5q/6K+IMqZL3gsM8m2MByXsL11ltu3f037/ZL1YEC0kZeTkw4i7RppvnQNQ
UrnmZRumxnPChVCuKxb4oCudaVutQ/nFZeyfAwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAC+7glCs
TKBSURkwmbA4k/SYVOOhhz3VkBX0he3r1/Vd6Qk8IlRJwQj5AT8e40gz+vqO0GvjaYAx70bvIGqn
yYE/VVJJOG5Zw6Tott69Dx4AOCrmbZb96zOAjlcEI3Ol7U1h+9+Uo2h5ZC8AMWnC3rk3VudrSB8d
AhBwZmY9l8AB </X509Certificate>
  </X509Data>
 </KeyInfo>
 <Object xmlns="" Id="value1">
  <value1 ID="1">
testval1</value1>
 </Object>
 <Object xmlns="" Id="value2">
  <value2 ID="2">
testval2</value2>
 </Object>
</Signature>
```

Example of Enveloping Signature with non-XML Input File

The following example shows the enveloping signature type with non-XML input file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
  <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <Reference URI="#sgconrad:-7cd5f978:1159315afbc:-6124">
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
   <DigestValue>
```

```
UWdFQL/JwoDirPg/AJdp+m5+bT4=</DigestValue>
  </Reference>
 </SignedInfo>
 <SignatureValue>
cAtHLlmNUVRmWzn4mDvWkcRvFDoKOkes+gMsnC4pHAKclg99j+e2xxR0SsE5HnvNPEH3IrwT
GZyaTXVlx3UTaX1C+215t0mW4CYn4nyZpwJTbM18pRZq8tijquydg4roZz/yawz856uow3KH
z+khzOuwt78GzwQXVyqQymyVrQk= </SignatureValue>
 <Object xmlns="" Encoding="base64" Id="sgconrad:-7cd5f978:1159315afbc:-6124">
dGhpcyBpcyB0ZXN0 </Object>
</Signature>
```

Example of Detached Signature

The following example shows the detached signature type:

```
<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
 <SignedInfo>
  <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <Reference URI="sgconrad:31e5343c:1158d3b080f:-75fc">
   <Transforms>
    <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
     <XPath>
descendant-or-self::Contract</XPath>
    </Transform>
   </Transforms>
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
   <DigestValue>
1ybLCHRnYSGKCoswkUOuD650Mr0=</DigestValue>
  </Reference>
  <Reference URI="sgconrad:-3c3ab664:1158cfc1d5d:-5431">
   <Transforms>
    <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
     <XPath>
descendant-or-self::FILLER</XPath>
    </Transform>
   </Transforms>
   <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
   <DigestValue>
2jmj7l5rSw0yVb/vlWAYkK/YBwk=</DigestValue>
  </Reference>
 </SignedInfo>
 <SignatureValue>
CBML9dFb/hEQXXR7oYfTuu4qit/VhUjwIfvPhSUQTQg0j+BFiTZFwNZaCJkZGswxDnSKhH1p
CuLn/Fpzl2CJpNduDU0Ff0pstd7MITS0l0/IvhDVS+Tf6WiYkN8UYTCkJeg063zlbW+15mR1
Z25jCs0gWO9qEStht34qXRi7ii0= </SignatureValue>
</Signature>
```

Example of Combination Signature of Enveloped/Enveloping/ Detached

The following example shows the combination signature type:

```
<?xml version="1.0" encoding="UTF-8"?>
<test>
 <value1 ID="1">
testval1</value1>
 <value2 ID="2">
testval2</value2>
 <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
   <CanonicalizationMethod
Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
   <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
   <Reference URI="#1">
```

```
    <Transforms>
     <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
IpBpovbT2WG7C+gTME1Np/V2fqo=</DigestValue>
   </Reference>
   <Reference URI="#2">
    <Transforms>
     <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
pFXQ3ZZv4Fivm2MFs6vpfEanEDI=</DigestValue>
   </Reference>
   <Reference URI="#sgconrad:-56000361:115d676b12e:-7988">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
7bPj9pPMJpsJw1OJ1b2jsrhxYMY=</DigestValue>
   </Reference>
   <Reference URI="sgconrad:-56000361:115d676b12e:-795f">
    <Transforms>
     <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
      <XPath>
descendant-or-self::node1</XPath>
     </Transform>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
2gFHdr03uDeDqwcxGveD+uYDIjM=</DigestValue>
   </Reference>
   <Reference URI="sgconrad:-56000361:115d676b12e:-795f">
    <Transforms>
     <Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
      <XPath>
descendant-or-self::node2</XPath>
     </Transform>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>
nn7t7PJs5RqDp1BKZ4j1BxhX2ik=</DigestValue>
   </Reference>
  </SignedInfo>
  <SignatureValue>
Ui7XYcZNkXG+9OqNgKrcRJoyCuhpoRtVmFnXVOMf8aAuGXZw3FwFxz7VLKv9c1K8ZUNW9vCs
G4EpahlCS4AcpbVBwv00HvkhA11/tqYYB9kRK/wM4cb6sN5ULbQ4Ab0j9xyFKOQ6sr2MawOx
fdNEes6XAHbpWZvxKDR4vWxAFnE= </SignatureValue>
  <Object xmlns="" Encoding="base64" Id="sgconrad:-56000361:115d676b12e:-7988">
dGhpcyBpcyBhbiBpbnZhbGlkIHRlc3QgZG9jIGZvciB4bWwgZHNpZyBzZXJ2aWNlLgo=</Object>
 </Signature>
</test>
```

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.

1623-14, Shimotsuruma, Yamato-shi

Kanagawa 242-8502 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 2011. Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. 2011.

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.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium and the Ultrium Logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Connect Control Center®, Connect:Direct®, Connect:Enterprise®, Gentran®, Gentran®:Basic®, Gentran:Control®, Gentran:Director®, Gentran:Plus®, Gentran:Realtime®, Gentran:Server®, Gentran:Viewpoint®, Sterling Commerce™, Sterling Information Broker®, and Sterling Integrator® are trademarks or registered trademarks of Sterling Commerce™, Inc., an IBM Company.

Other company, product, and service names may be trademarks or service marks of others.

Product Number:

Printed in USA