WebSphere®

# IBM WebSphere Partner Gateway Enterprise and Advanced Editions

Version 6.2.1

# *Hub Configuration Guide*

**Note**
Before using this information and the product it supports, read the information in "Notices" on page 437.

**February 2011**

This edition applies to version 6, release 2, modification 1 of IBM WebSphere Partner Gateway Enterprise Edition (product number 5724-L69) and version 6, release 2, modification 1 of Advanced Edition (product number 5724-L68) and to all subsequent releases and modifications until otherwise indicated in new editions.

# Contents

# Chapter 1. About this book

This document describes how to configure the IBM (R) WebSphere (R) Partner Gateway server.

## Audience

Administrators maintain WebSphere Partner Gateway. This book assumes two types of administrators:

- Hub administrator
- Account administrator

The hub administrator is the super user in the community. The hub administrator is responsible for overall hub community configuration and management, including partner configuration and connection activation. The account administrator has access to a subset of the hub administrator features and is the main administrative user for the internal partner or external partner.

**Note:** The console of Hub administrator, External Partners, and Internal Partners will be different depending on their access controls/rights.

## Typographic conventions

This document uses the following conventions.

*Table 1. Typographic conventions*

| Convention | Description |
|---|---|
| `Monospace font` | Text in this font indicates text that you type, values for arguments or command options, examples and code examples, or information that the system prints on the screen (message text or prompts). |
| **bold** | Boldface text indicates graphical user interface controls (for example, online button names, menu names, or menu options) and column headings in tables and text. |
| *italics* | Text in italics indicates emphasis, book titles, new terms and terms that are defined in the text, variable names, or letters of the alphabet used as letters. |
| *`Italic monospace font`* | Text in italic monospace font indicates variable names within monospace-font text. |
| *ProductDir* | *ProductDir* represents the directory where the product is installed. All IBM WebSphere Partner Gateway product path names are relative to the directory where the IBM WebSphere Partner Gateway product is installed on your system. |
| *`%text%`* and *`$text`* | Text within percent signs (%) indicates the value of the Windows(R) `text` system variable or user variable. The equivalent notation in a UNIX(R) environment is $*text*, indicating the value of the *text* UNIX environment variable. |
| Underlined colored text | Underlined colored text indicates a cross-reference. Click the text to go to the object of the reference. |

*Table 1. Typographic conventions (continued)*

| Convention | Description |
|---|---|
| Text in a blue outline | (In PDF files only) An outline around text indicates a cross-reference. Click the outlined text to go to the object of the reference. This convention is the equivalent for PDF files of the "Underlined colored text" convention included in this table. |
| " " (quotation marks) | (In PDF files only) Quotation marks surround cross-references to other sections of the document. |
| { } | In a syntax line, curly braces surround a set of options from which you must choose one and only one. |
| [ ] | In a syntax line, square brackets surround optional parameters. |
| < > | Angle brackets surround variable elements of a name to distinguish them from one another. For example, <*server_name*><*connector_name*>tmp.log. |
| / or \ | Backslashes (\) are used as separators in directory paths in Windows installations. For UNIX installations, substitute slashes (/) for backslashes. |

## Related documents

The complete set of documentation available with this product includes comprehensive information about installing, configuring, administering, and using WebSphere Partner Gateway Enterprise and Advanced Editions.

You can download the documentation or read it directly online at the following site:

http://www.ibm.com/software/integration/wspartnergateway/library/

**Note:** Important information about this product may be available in Technical Support Technotes and Flashes issued after this document was published. These can be found on the WebSphere Business Integration Support Web site:

http://www.ibm.com/software/integration/wspartnergateway/support/

Select the component area of interest and browse the Technotes and Flashes section.

## New in release 6.2.1

WebSphere Partner Gateway V6.2.1 supports the following new features:

- Web Mail box is web based support for B2B interaction. Partners, customers, and vendors interact with the WebSphere Partner Gateway hub using only the internet browser.
- SFTP integrated server is supported in addition to FTP integrated server.
- OpenPGP certificate is supported in WebSphere Partner Gateway.
- Support for WebSphere Application Server ND V7.0.0.13, WebSphere Messaging Queue 7.0, and WTX 8.3.
- Platform support for Windows 2008, Windows 7, and SLES 11.

- Power 7 Support -Toleration Mode (P6/P6+ Compatible Modes).
- Virtualization Support - VMware® ESX with Windows and Linux, Power VM with AIX.

# Chapter 2. Introduction to hub configuration

After you install WebSphere Partner Gateway and before any documents can be exchanged between the internal partners and external partners, you must configure the WebSphere Partner Gateway server (the hub).

This chapter covers the following topics:



## Overview of hub configuration

The goal is to enable the internal partner to send a document or set of documents (electronically) to an external partner or to receive a document or set of documents from an external partner. The hub manages the receipt of the documents, the transformation to other formats (if required), and the delivery of the documents. The hub can also be configured to provide security for incoming and outgoing documents.

The documents exchanged between the hub and a partner are typically in a standard format and represent a specific business interaction. For example, a partner might send a purchase order request as a RosettaNet 3A4 PIP, a cXML OrderRequest document, or an EDI-X12 interchange with an 850 transaction. The hub transforms the document into a format that can be used by an application at the internal partner. Similarly, an internal partner back-end application might send a purchase order response in its own custom format that is transformed into a standard format. The transformed document is then sent to the partner.

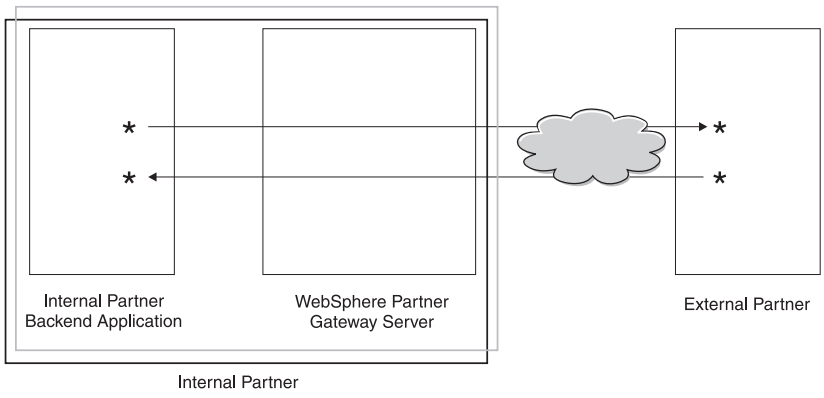



*Figure 1. How documents flow through the hub*

In this guide, you will see how to configure the hub and then how to set up the partners. You will also learn how to configure security for the hub.

Notice in Figure 1 that the WebSphere Partner Gateway server and the internal partner back-end application are all owned by the internal partner. The internal

partner is the company that owns the hub. As you will see in later chapters, you define a profile for the internal partners just as you do for external partners.

**Note:** This document shows you how to create connections that flow from the internal partner back-end application to a partner destination and from an external partner to the internal partner destination. After the documents arrive at the internal partner destination, you will probably want to integrate them with a back-end application, such as WebSphere InterChange Server or WebSphere MQ Broker. The tasks required to integrate between WebSphere Partner Gateway and such back-end applications are defined in the *WebSphere Partner Gateway Enterprise Integration Guide*.

## Information needed to set up the hub

You need some information about the types of exchanges in which the internal partner will participate in order to set up the hub. For example, you need the following information:

- Which types of documents (for example, EDI-X12 or custom XML) will the internal partners and its external partners be sending through the hub?
- Which types of transports (for example, HTTP or FTP) will the internal partners and its external partners use to send the documents?
- Will a document coming into the hub need to be split into multiple documents, or will individual documents coming into the hub need to be grouped before being sent on?
- Will the documents be transformed before being delivered?
- Will the documents be validated before being delivered?
- Will a document be checked to see if it is a duplicate before being delivered?
- Will the documents be encrypted or digitally signed or use some other security technique?

When this information is determined, you are ready to begin setting up the hub.

After you define the hub, you can define your external partners, using information (such as IP address and DUNS numbers) that is supplied to you by the external partners. As noted earlier, you also define the internal partner as a special type of partner of the hub.

### Overview of transports

Documents can be sent from partners to the WebSphere Partner Gateway (the hub) over a variety of transports. A partner can send documents over public networks using HTTP, HTTPS, JMS, FTP, FTPS, FTP Scripting, SMTP, SFTP, or a file directory. A partner can send documents over a Value Added Network (VAN), a private network, using the FTP Scripting Transport. You can create your own transport, as well.

**Note:** When the file-directory transport is used between a partner and the hub, the administrator should take care of all the security-related issues.

Similarly, the hub sends documents to back-end applications over a variety of transports. The most commonly used transports between the hub and back-end applications are HTTP, HTTPS, JMS, File-directory, FTP Scripting, FTP, SFTP, and SMTP.

Figure 2 shows the HTTP, HTTPS, JMS, and file-directory transports.

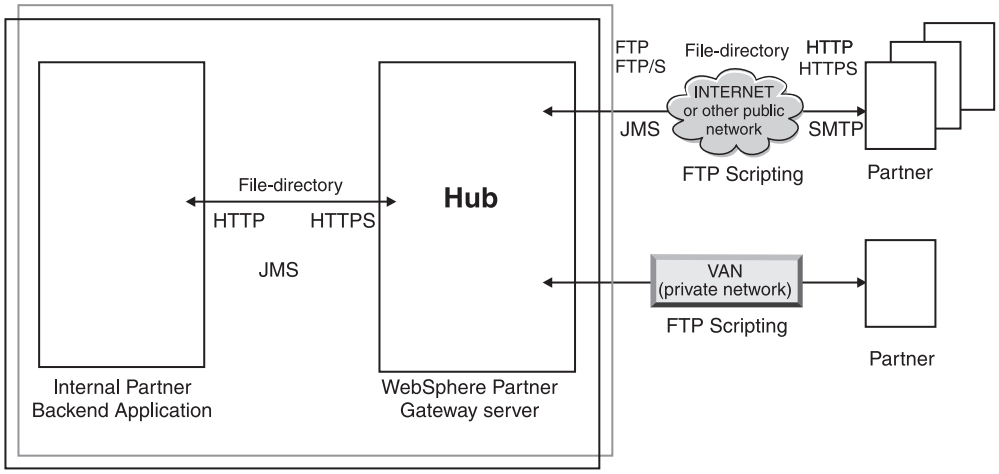



*Figure 2. Most common transports supported by WebSphere Partner Gateway*

The type of transport used to send and receive documents affects the setup of receivers and destinations. A *receiver* is an entry point into the hub--the place where documents sent by partners or back-end applications are received at the hub. A *destination* is an entry point into the partner's computer or the back-end system--the place where the hub sends documents. To prepare to use the FTP, FTPS, FTP Scripting, JMS, and file-directory transports, you have to do some setup work, as described in Chapter 4, “Preparing to configure the hub,” on page 33.

## Overview of document definitions

When you set up the interchange of documents between external partners and internal partners, you specify several things about the document:

- The *packaging* that surrounds the document
- The business *protocol* that defines a class of documents that share some common characteristics
- The *document type* that identifies one of the documents that are provided by the business protocol

The packaging of the document, the protocol of the document, and the document type make up the *document definition*. Suppose you use the product-provided document definition of:

- Packaging: AS
- Protocol: EDI-X12
- Document type: ISA

This is what happens when a document is received that conforms to this routing definition. After the hub receives the document, the fixed inbound workflow unpackaging step determines that AS packaging is used by the document. This is because of the presence of transport headers that are specified for AS packaging. Other packaging types are discovered by the hub in a similar manner, typically by examining transport headers that came in with the document. When there is not a match with any packaging type, the None packaging type is assigned to the document. In the case of AS packaging, the from and to business identifiers are

obtained from the message transport headers. Also carried in the AS transport headers are other headers that can specify whether or not the message is encrypted, compressed, or signed.

After identifying the packaging, the hub fixed inbound workflow protocol parse step determines the protocol and document type of the document. This is done by examining the actual message content and looking for characteristics in the document that identify the protocol and document type. The protocol parse workflow step also extracts other information from the document depending on the protocol that is used.

Once the document is known to use a particular package, protocol, and document type, the hub can proceed with processing the document. At this point it will know the from and to business ids in addition to the package, protocol, and document type. Given this information, the hub can search for a Connection between the from and to partners that has the inbound package, protocol, and document type.

Once the Connection is found, the hub knows how to route and process the document because it can find the following additional information:

- Certificates for the from and to partners (if required)
- Attribute settings for the from-routing and the to-routing
- The Action to perform when routing the document
- The applicable transformation map (if any)
- The applicable validation map (if any)

## Packaging

The packaging provides information that pertains to the transmission of the document. As mentioned in the previous section, if the packaging is AS, the hub uses information in the AS header to determine the source and destination for the document. If a partner is sending a RosettaNet PIP to the internal partner, the PIP is packaged as RNIF.

Figure 3 shows you the packaging types that can be set for documents exchanged between the hub and an external partner and between the hub and a back-end application.

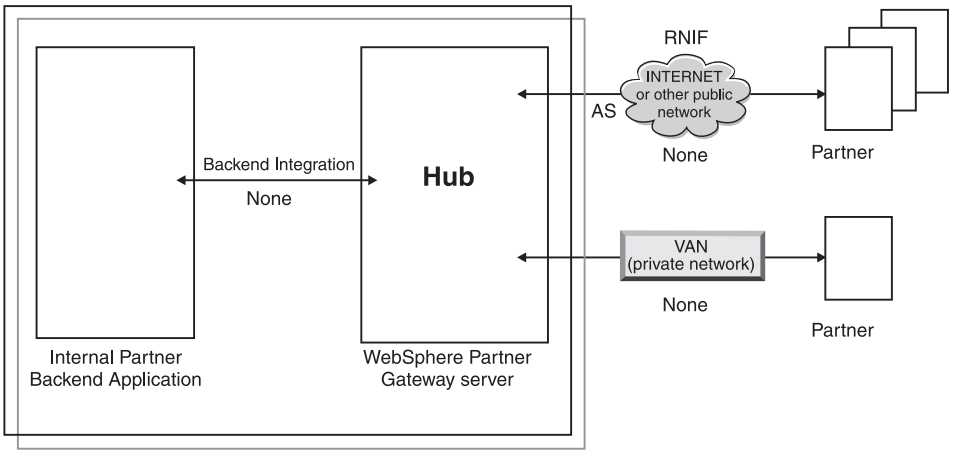



*Figure 3. Document packaging types*

Packages are associated with specific protocols. For example, a partner must specify RNIF packaging when sending a RosettaNet document to the hub.

**Backend Integration:** As shown in Figure 3 on page 8, Backend Integration is available only between the hub and the back-end application. When you specify Backend Integration packaging, documents sent by the hub to the back-end system have special header information added. Similarly, when a back-end application sends documents with a packaging of Backend Integration to the hub, it must add header information. The Backend Integration package, and the requirements for the header information, are described in the *WebSphere Partner Gateway Enterprise Integration Guide*.

**AS:** The AS package is most commonly used between partners and the hub. The AS package can be used for documents that adhere to the AS1, AS2, and AS3 standards. AS1 is a standard used for securely transmitting documents over SMTP, and AS2 is a standard used for securely transmitting documents over HTTP or HTTPS. AS3 is a new standard used for securely transmitting documents over FTP or FTPS. Documents sent by a partner with a packaging of AS have AS1, AS2, or AS3 header information. Documents sent to a partner that expects AS1, AS2, or AS3 headers must be packaged (at the hub) as AS.

**None:** The None package can be used to send and receive documents between the hub and partners and between the hub and a back-end application. No header information is added (or expected) when a document is packaged as None.

**RNIF:** The RNIF package is provided on the installation medium. You upload the RNIF package (along with any PIPs you want exchanged), as described in "RosettaNet documents" on page 105. The RNIF package is used to send RosettaNet documents from the partner to the hub or from the hub to the partner.

**ebMS:** The ebXML Message Service (ebMS) mechanism provides a standard way to exchange business Messages among ebXML Trading Partners. It provides a reliable means to exchange business messages without relying on proprietary technologies and solutions. An ebXML message contains structures for a message header (necessary for routing and delivery) and a payload section.

ebMS provides a standard way to exchange business Messages among ebXML Trading Partners. An ebXML message is a communication protocol independent MIME/Multipart message envelope.

**N/A:** Some document types either end in WebSphere Partner Gateway or originate internally from WebSphere Partner Gateway. For document types ending in WebSphere Partner Gateway, no packaging is required. Document types originating internally from WebSphere Partner Gateway do not have source packaging. Therefore, for such flows, the packaging is specified as N/A.

For most one-way transmissions between an external partner and the internal partner (or vice versa), WebSphere Partner Gateway receives a document from an external partner and sends it to the internal partner. In WebSphere Partner Gateway, when creating the partner connection, you specify the packaging in which WebSphere Partner Gateway will receive the document and the packaging WebSphere Partner Gateway will use to send the document. In Figure 4 on page 10, a document packaged as AS is flowing from an external partner to the internal partner back-end. The document is delivered to the internal partner destination with no transport headers. In Figure 4 on page 10, one activity is associated with the exchange of documents.

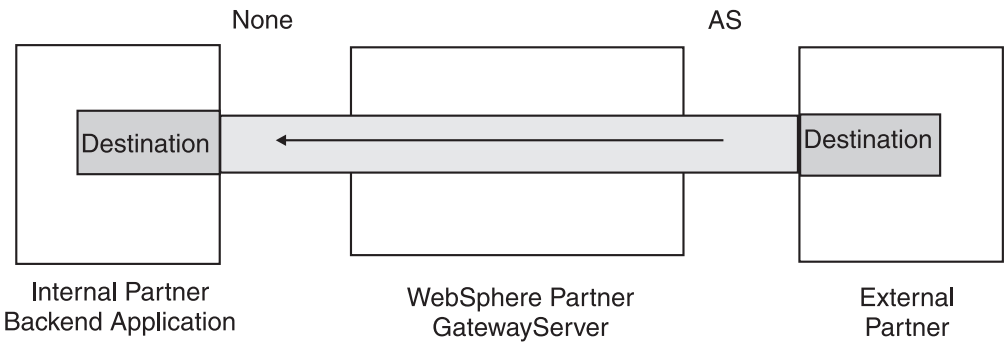



*Figure 4. Typical one-way connection*

Certain protocols, however, involve multiple activities (such as de-enveloping and transformation), some of which occur as intermediate parts of the overall exchange. For example, if a partner sends an EDI interchange to the hub, for eventual delivery to the internal partner, the interchange is de-enveloped and the individual EDI transactions are processed. The original EDI interchange has a package associated with it when it is sent from the partner. However, because the interchange itself is not delivered to the internal partner (it is de-enveloped within the hub and no additional processing of the interchange occurs), packaging of the interchange does not apply. When you set up the interaction for the de-enveloping step, therefore, you enter a package on the sending side but you specify N/A for the receiving side.

The process for setting up the document definitions required for an EDI exchange is described in Chapter 10, “Configuring EDI document flows,” on page 161.

## Protocols

The protocols that are provided with the system are:

- Binary

  The Binary protocol can be used with AS, None, and Backend Integration packages. A binary document contains no data about the source or destination of the document.

- EDI-X12, EDI-Consent, EDI-FACT

  These EDI protocols can be used with the AS or None packages. As described in “ N/A” on page 9, if the EDI transaction or interchange originates from the hub or ends at the hub, you specify N/A for the package. X12 and EDIFACT are EDI standards used for the exchange of data. EDI-Consent refers to content types that are specified in the EDI-Consent specification.

- Web Service

  Web service requests can be used only with the None package.

- cXML

  cXML documents can be used only with the None package.

- XMLEvent

  XMLEvent is a special protocol used to provide event notification for documents flowing to and from a back-end application. It can be used only with the Backend Integration package. This protocol is described in the *WebSphere Partner Gateway Enterprise Integration Guide*.

When you upload RNIF packages, you also get the associated protocols (RosettaNet and RNSC). RosettaNet (which is the protocol used between the partner and the hub) is associated with the RNIF package. RNSC (which is the

protocol used between the hub and the internal partner back-end application) is associated with the Backend Integration package.

For transformation of EDI transactions or XML or ROD documents, Data Interchange Services client (DIS) or WTX design studio is used to create transformation maps.

In the Data Interchange Services client, dictionaries are defined for the protocol associated with this transformation. A dictionary contains information about all of the EDI document definitions, segments, composite data elements, and data elements that make up the EDI Standard. The definitions of the source documents for EDI is supplied by WDI, whereas for ROD and XML, you need to create in DIS client. From Version 6.2.1 onwards, the standard and transformation maps can be compiled separately. For detailed information about a particular EDI Standard, consult the appropriate EDI Standards manuals. For information about the Data Interchange Services client, refer to the *WebSphere Partner Gateway Mapping Guide* or to the online help provided with the Data Interchange Services client.

**Note:** The sender and receiver IDs must be part of the ROD document definition associated with the transformation map. The information necessary to determine the document type and dictionary values must also be present in the document definition. Make sure that the Data Interchange Services client mapping specialist is aware of these requirements when creating the transformation map.

You can create custom protocols to define exactly how you want a document to be structured. For XML documents, you can define an XML format, as described in "Custom XML document processing" on page 148.

## Document type

The document itself can be in a variety of formats. The product-provided document types and their associated protocols are:

- Binary, which can be used with the Binary protocol
- ISA, which represents the X12 interchange (envelope) and which is associated with the EDI-X12 protocol
- BG, which represents the EDI Consent envelope and which is associated with the EDI-Consent protocol
- UNB, which represents the EDIFACT envelope and which is associated with the EDI-EDIFACT protocol
- XMLEvent, which can be used with the XMLEvent protocol

The following list describes other types of documents and the source of their definition:

- A RosettaNet PIP (which you upload from the installation medium), which can be used with the RosettaNet protocol
- A Web service (which you upload as a WSDL file), which can be used with the Web Service protocol
- A cXML document (which you create by specifying the type of cXML document)
- A specific EDI standard transaction, which you import from the Data Interchange Services client
- A record-oriented-data (ROD) or XML document, which you import from the Data Interchange Services client

You can also create your own document types, as described in "Custom XML document processing" on page 148.

## Overview of document processing

Before you begin setting up the hub, it is helpful to review the components of WebSphere Partner Gateway and how they are used to process documents.

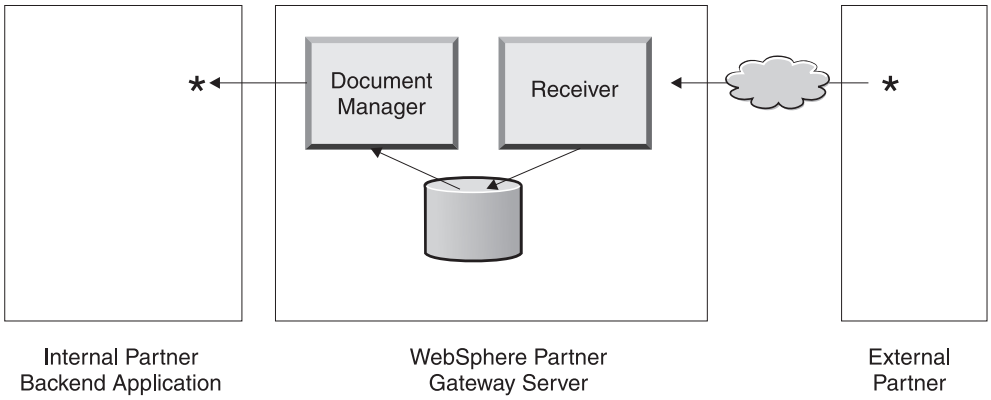



*Figure 5. The Receiver and Document Manager components*

Figure 5 is an example of how a document is sent from a partner, received at the hub, processed at the hub, and sent to an internal partner back-end application.

**Note:** For purposes of illustration, the drawings in this document show one Receiver component and one Document Manager, installed on the same server machine. (Not shown is the third component, the Console, which is the interface to WebSphere Partner Gateway.) You can, in fact, have multiple occurrences of these components, and they can be installed on different servers. All components must use the same common file system. See the *WebSphere Partner Gateway Installation Guide* for information about the different topologies that can be used to set up WebSphere Partner Gateway.

A document is received into WebSphere Partner Gateway by the Receiver component. The Receiver component is responsible for monitoring transports for inbound documents, retrieving the documents that arrive, performing some basic processing on them, and then queueing them so that the Document Manager can retrieve them.

Receivers instances are transport-specific. You set up a receiver for each type of transport the hub will support. For example, if partners are going to send documents over HTTP, you set up an HTTP receiver to receive them.

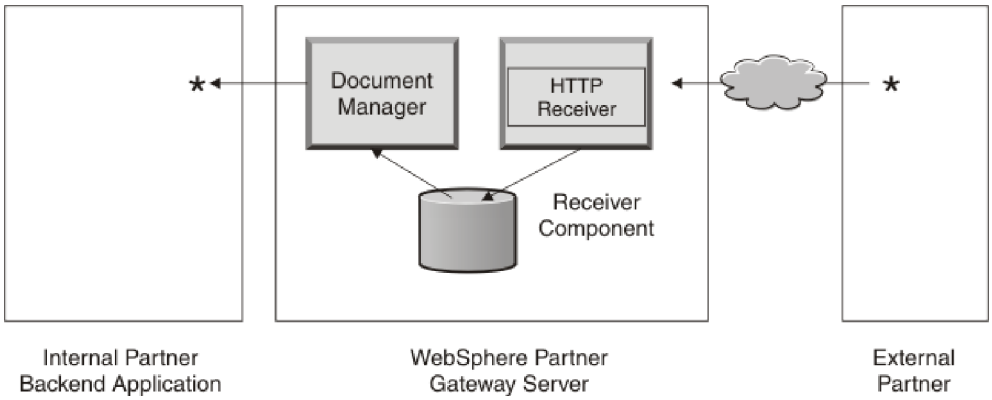



*Figure 6. An HTTP receiver*

If the internal partner back-end application is going to send documents over JMS, you set up a JMS receiver at the hub to receive them.

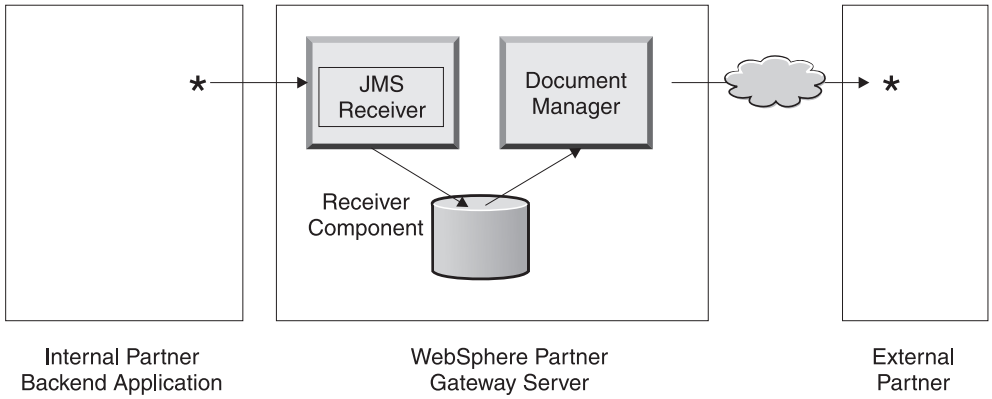



*Figure 7. A JMS receiver*

As described in "Overview of transports" on page 6, WebSphere Partner Gateway supports a variety of transports, but you can also upload your own user-defined transport to define a receiver (as described in " Setting up a receiver for a user-defined transport" on page 72).

The Receiver component sends the document to a shared file system. For multiple documents that are in a single file (for example, XML or ROD documents or EDI interchanges sent together), the receiver splits the documents or interchanges before sending them to the shared file system. The Document Manager component retrieves the document from the file system and determines the routing information and whether any transformation is required.

For example, the internal partner might send an EDI-X12 document with None packaging to the hub, for delivery to a partner that is expecting the EDI-X12 document with AS2 packaging. The partner provides the HTTP URL where the AS2-packaged document should be delivered, and the Document Manager packages the document as expected by the partner. The Document Manager uses the configuration of the destination for that partner (which must have been set up for the HTTP URL where the partner expects to receive AS2 documents) to send the document to the partner.

# Configuring document processing components with handlers

This section describes, in more detail, the components of WebSphere Partner Gateway and shows you the various points at which you can (or must) change the product-provided behavior of the components for processing a business document.

You use *handlers* to change the product-provided behavior of receivers, destinations, fixed workflow steps, and actions. There are two types of handlers--those supplied by WebSphere Partner Gateway and those that are user-defined. See the *WebSphere Partner Gateway Programmer Guide* if you want information about creating handlers.

After a handler is created, you upload it to make it available. You upload only user-defined handlers. The handlers supplied by WebSphere Partner Gateway are already available.

The sections that follow describe the processing points at which you can specify handlers.

## Receivers

Receivers have three *configuration points* for which handlers can be specified--Preprocess, SyncCheck, and Postprocess.

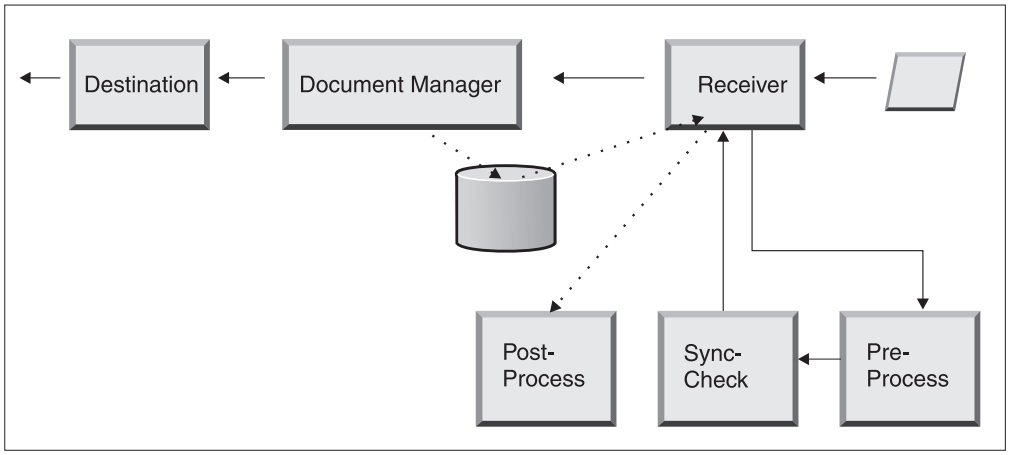



*Figure 8. Receiver configuration points*

The processing occurs in the following order:

1. The Receiver component calls the Preprocess and SyncCheck steps after it receives the document.
2. It then calls the Document Manager to process the document.
3. In the case of synchronous flows, the Document Manager provides a Sync Response. The Receiver component then calls the Postprocess step with the response returned from the Document Manager.

The steps are described in the following sections:

- Preprocess

  The Preprocess step is generally used for any processing on the document that needs to be accomplished before the document can be processed by the Document Manager. For example, if you will be receiving multiple ROD documents in a single file, you configure the ROD splitter handler when you

define the receiver. The ROD splitter, along with two other product-provided splitters, are available for you to use when you set up a receiver. If you create additional handlers for the preprocess step, those handlers are also available.

See " Preprocess" on page 72 for information about configuring the Preprocess configuration point.

- SyncCheck

  SyncCheck is used to determine whether WebSphere Partner Gateway should process the document synchronously or asynchronously. For example, in the case of AS2 documents received over HTTP, it determines whether an MDN (message disposition notification) should be returned synchronously over the same HTTP connection. WebSphere Partner Gateway supplies a variety of handlers for synchronous checking. The list of handlers varies, depending on the transport associated with the receiver.

  SyncCheck applies only to those transports (such as HTTP, HTTPS, and JMS) that support synchronous transmission.

  **Note:** For AS2, cXML, RNIF, or SOAP documents that will be used in synchronous exchanges, you must specify the associated SyncCheck handler on the HTTP or HTTPS receiver.

  See " SyncCheck" on page 76 for information about configuring the SyncCheck configuration point.

- Postprocess

  Postprocessing is used for processing the response document that the hub sends as the result of a synchronous transaction.

  See " Postprocess" on page 77 for information about configuring the Postprocess configuration point.

## Document Manager

Documents received by receivers are picked up by the Document Manager from the common file system for further processing. The Document Manager uses partner connections to route the documents. All documents flowing through the Document Manager go through a series of workflows: fixed inbound workflow, variable workflow, and fixed outbound workflow. At the end of the inbound workflow, the partner connection is determined. The partner connection specifies the action to perform on this document. After executing the variable workflow, the Document Manager processes the fixed outbound workflow on this document.

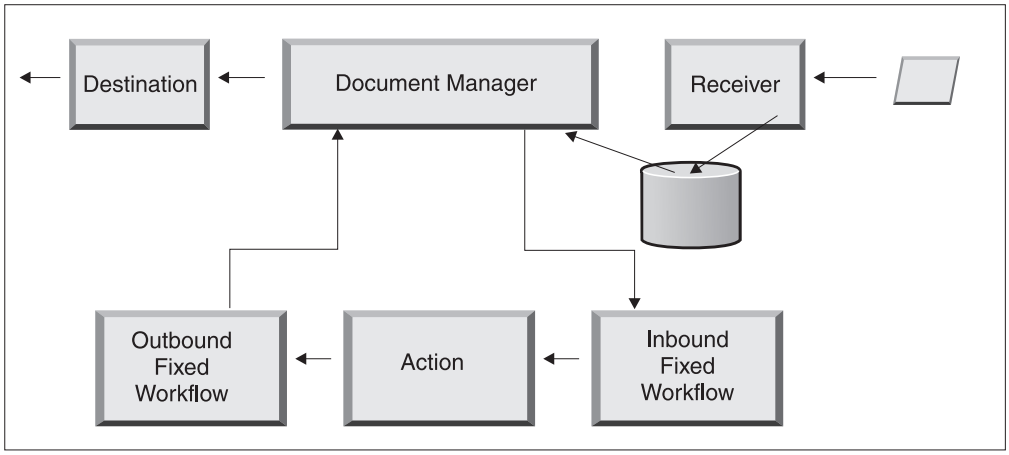



*Figure 9. Fixed workflows and actions*

Figure 9 shows the path that a document such as a RosettaNet PIP or a Web service would take. Some documents, however, require several configured flows. For example, an EDI interchange can consist of multiple transactions. The first flow uses an action to de-envelope the set of individual transactions. Each of these transactions is reintroduced and processed in its own configured flow.

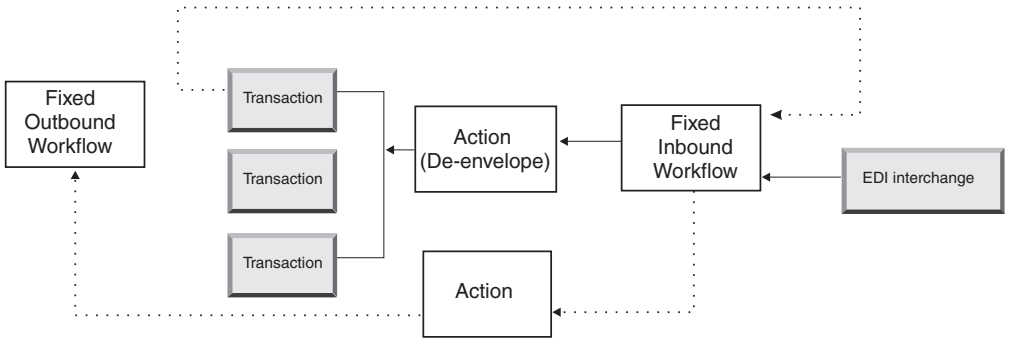



*Figure 10. Fixed workflows and actions for an EDI interchange*

## Inbound fixed workflow

The Inbound fixed workflow consists of the standard set of processing steps performed on all documents coming into the Document Manager from a Receiver. The workflow is fixed because the number and types of steps are always the same. Through user exits, however, you can provide customized handlers for processing the following steps: Protocol Unpackaging and Protocol Processing. The last step of inbound fixed workflow performs partner connection lookup, which determines the variable workflow that runs for this business document.

For example, if an AS2 message is received, the message is decrypted, and the sender and receiver business IDs are retrieved. The inbound fixed workflow steps convert the AS2 document into plain text for further processing by WebSphere Partner Gateway and extract information to determine the action for the message.

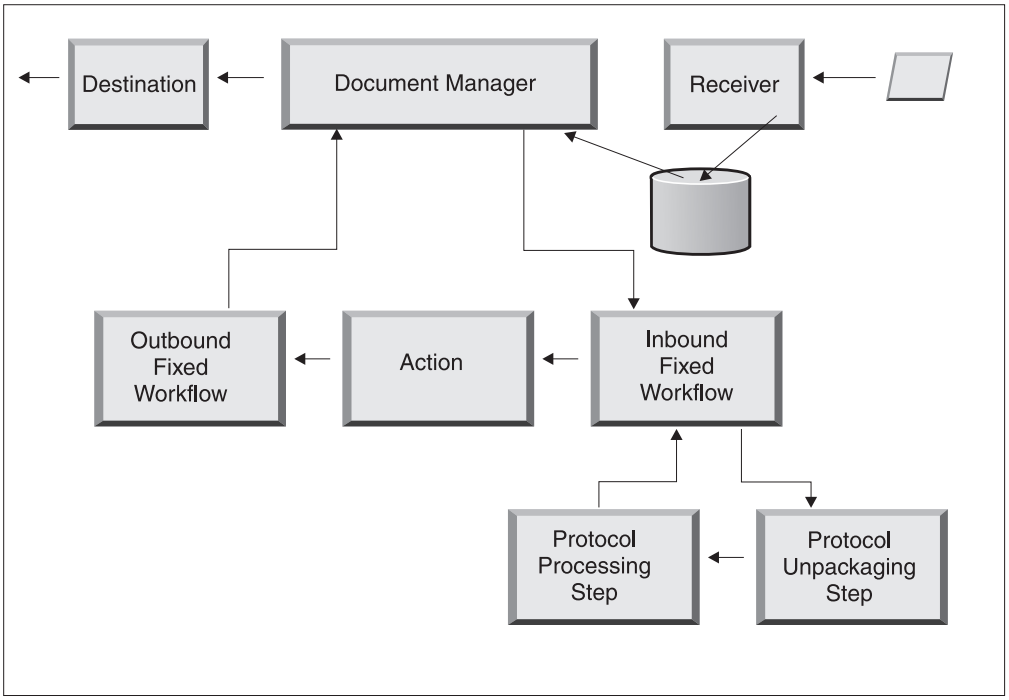



*Figure 11. Inbound fixed workflow steps*

**Protocol Unpackaging:** During Protocol Unpackaging, a document is unpackaged so that it can be further processed. This process can include decryption, decompression, signature verification, extraction of routing information, user authentication, or business document parts extraction.

WebSphere Partner Gateway provides handlers for RNIF, AS, Backend Integration, and None packaging. If handlers for other packaging protocols are necessary, they can be developed as user exits. Refer to the *WebSphere Partner Gateway Programmer Guide* for information about writing user exits.

You cannot modify the Protocol Unpackaging step; however, you can add business logic to the step by adding handlers.

See “Configuring fixed workflows” on page 80 for information about configuring this step.

**Protocol Processing step:** Protocol Processing involves determining protocol-specific information, which might include parsing the message to determine routing information (such as the sender ID and the receiver ID), protocol information, and document type information. WebSphere Partner Gateway provides processing for a variety of protocols, as listed in “ Protocol processing handlers” on page 81. Processing for other protocols—for example, CSV (comma-separated value)—can be provided with a user exit.

You cannot modify the Protocol Processing step; however, you can add business logic to the step by adding handlers.

See “Configuring fixed workflows” on page 80 for information about configuring this step.

You can use the default handler that applies to the protocol for your document, or you can specify a different handler for the Protocol Unpackaging and Protocol Processing fixed workflow steps.

## Actions

The next step in the processing sequence occurs based on the actions that have been set up for the document exchange. Actions consist of a variable number of steps that can be performed on the document. Examples of actions are validation of a document (so that it conforms to a particular set of rules) and transformation of the document to the format required by the recipient.

If the document has no specific steps required, it can use the product-provided Pass Through action, which makes no changes to the document.

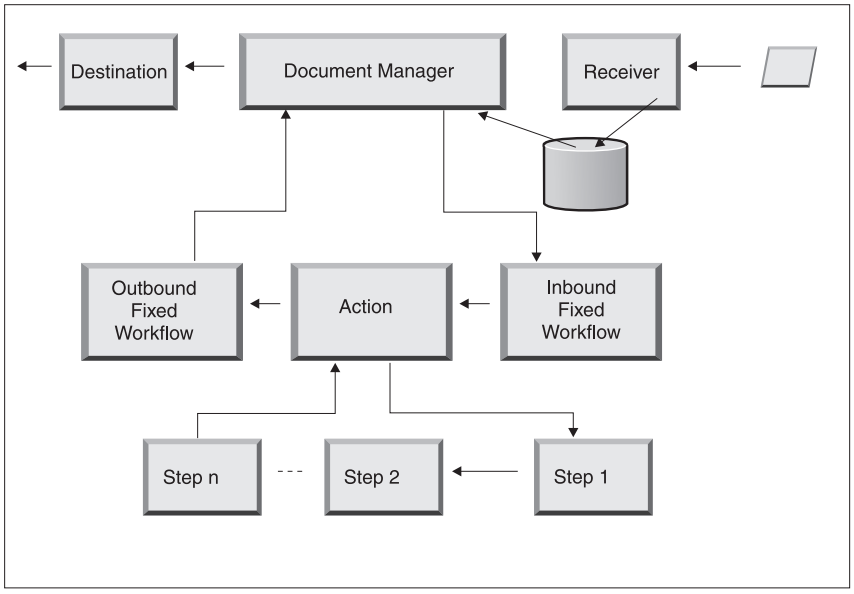



*Figure 12. Action steps*

You cannot modify a product-provided action. You can, however, create an action (and add handlers to the configured list) or copy a product-provided action and then modify the list of handlers.

See "Configuring actions" on page 81 for information about creating or copying a product-provided action or configuring a user-defined action.

**Related concepts**

"Configuring actions" on page 81

## Outbound fixed workflow

The Outbound Fixed Workflow consists of one step—the packaging of the document with its protocol information. For example, if a document has been set up to be received by a back-end application using Backend Integration packaging, certain header information is added to the document before it is passed to the destination.

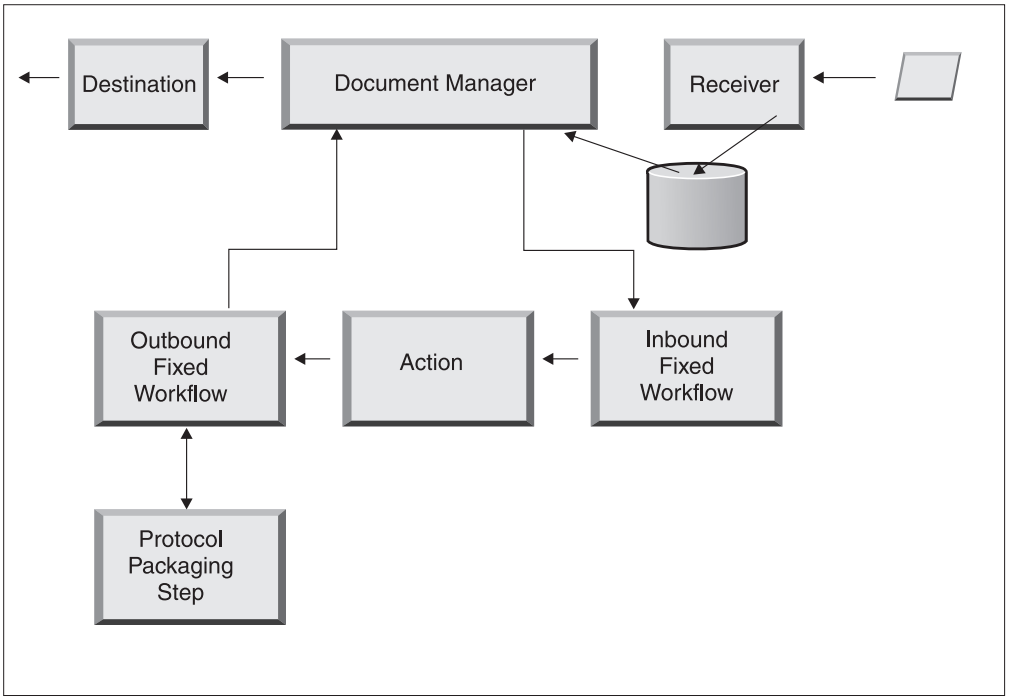



*Figure 13. Outbound fixed workflow steps*

WebSphere Partner Gateway provides handlers for a variety of packages and protocols, as listed in "Outbound workflow" on page 81. If other packaging handlers are required, they can be developed as user exit steps. Typically these steps take care of one or more of the following processes:

- Assembling or enveloping
- Encrypting
- Signing
- Compressing
- Setting business-protocol-specific transport headers

You cannot modify the Protocol Packaging step; however, you can add business logic to the step by adding handlers.

See "Configuring fixed workflows" on page 80 for information about configuring this workflow step.

## Destinations

Destinations are configured in the console for each partner to which you need to send messages. The configuration of a destination includes the transport that will be used to send messages and the configuration needed to send it such as the URL for the partner's receiving process.

After the document leaves the Document Manager, it is sent using a destination to the intended recipient. The destination has two configuration points—Preprocess and Postprocess.

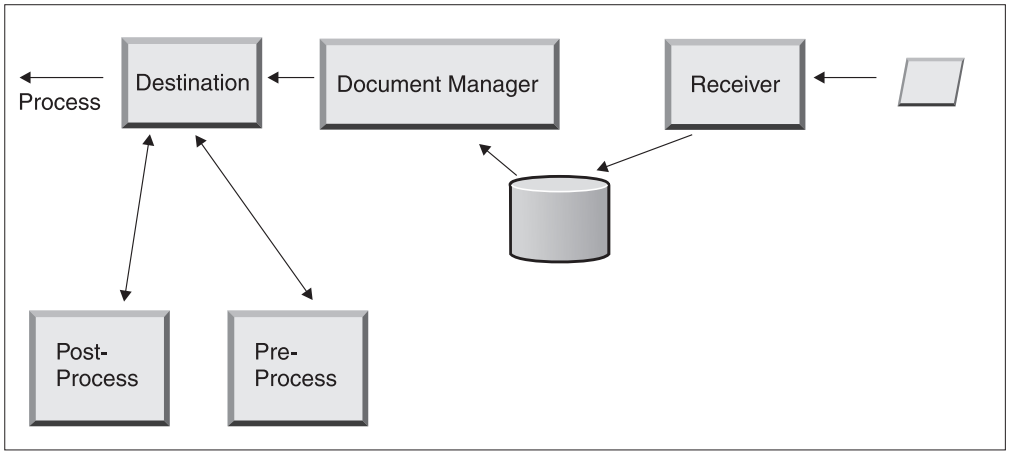



*Figure 14. Destination configuration points*

- Preprocess

  Preprocess affects the processing of a document before it is sent to the recipient. (Process is the actual sending of the document.) No handlers are supplied by the system to configure the Preprocess step; however, you can upload a user-defined handler.
- Postprocess

  Postprocess acts on the results of the document transmission (for example, on the response it receives from the recipient during a synchronous transmission). No handlers are supplied by the system to configure the Postprocess step; however, you can upload a user-defined handler.

See "Configuring handlers" on page 228 for information about configuring the Preprocess and Postprocess steps.

# Overview of configuring the hub

After you have analyzed your business needs, as described in "Information needed to set up the hub" on page 6, you set up the hub and create your partner profiles. This section provides a high-level overview of the tasks involved.

**Note:** As you are configuring the hub, refer to the *WebSphere Partner Gateway Administrator Guide* for information on event codes and for troubleshooting tips.

## Setting up the hub

### About this task

As the hub administrator, you perform the following tasks to set up the hub:

1. Perform any preliminary setup (if required) for the transports you are using. The preliminary setup is described in Chapter 4, "Preparing to configure the hub," on page 33.
2. Optionally, customize the console and change the default password and permissions policy. These tasks are described in Chapter 6, "Configuring the Community Console," on page 49.
3. Create receivers for the types of transports that will be used to receive documents at the hub (from the internal partner and from external partners). Creating receivers is described in Chapter 7, "Defining receivers," on page 55.

**Note:** If you will be configuring the receiver with user-defined handlers, you must upload the handlers before creating the receiver. Uploading handlers is described in " Uploading user-defined handlers" on page 56.

4. Configure any inbound workflow steps or actions. This is an *optional* step and is needed only by those who have specific requirements for document processing not provided by WebSphere Partner Gateway. If you do not need to change the product-provided behavior of workflows or actions, skip this step. Configuring workflow steps and actions is described in Chapter 8, "Configuring fixed workflow steps and actions," on page 79.

   **Note:** You must upload the user-defined handlers before configuring workflows or actions. Uploading user-defined handlers is described in " Uploading handlers" on page 79.
5. Create document definitions (or verify that the ones you need are already available) to define the types of documents you can send or receive at the hub.
6. Create interactions to indicate the valid combination of two document definitions.

   Creating document definitions and creating interactions are described in Chapter 9, "Configuring document types," on page 99 and Chapter 10, "Configuring EDI document flows," on page 161.
7. Create a profile for the internal partner, providing information about the internal partner and establishing the types of documents that the internal partner can send and receive (the B2B capabilities of the internal partner). Creating the profile is described in Chapter 3, "Creating and setting up partners," on page 23.

## Creating partners

After you set up the hub, you create a profile for each external partner that will be exchanging documents with the internal partner. Only the hub administrator can create partners.

As the hub administrator, you can also set up the B2B capabilities of partners, establish the destinations for partners, and set up security profiles for partners. These steps can alternatively be performed by the partners themselves.

Creating partners is described in Chapter 3, "Creating and setting up partners," on page 23. Creating destinations is described in Chapter 11, "Creating destinations," on page 209. Setting up security profiles is described in Chapter 13, "Enabling security for document exchanges," on page 235.

## Establishing document connections

After you configure the hub and create partner profiles, you are ready to set up connections. Connections indicate the valid combinations of senders and receivers and the documents they can exchange. Managing connections is described in Chapter 12, "Managing connections," on page 231.

# Overview of OpenPGP Certificates

OpenPGP is supported in WebSphere Partner Gateway. It uses a combination of strong public-key and symmetric cryptography to provide security services. The various functions of OpenPGP, which are included in this release, are as follows:

- Packaged messages in accordance to RFC 4880.

**Note:** RFC 2440 and RFC 3156 are not supported in this release.

- Encryption, Encryption with modification detection, and Compression.

  **Note:** In this release, WebSphere Partner Gateway does not support signing using OpenPGP.
- Supported encryption algorithms are CAST5 (128 bit key), TripleDES (168 bit key), Blowfish (128 bit key), Twofish (256 bit key), AES (128, 192 & 256 bit key).

  **Note:** Twofish, TripleDES, and AES (192 & 256 bit key) require unrestricted cryptography jurisdiction policy files.
- Supported compression algorithms are ZIP, ZLIB, and BZip2.
- ASCII Armored messages.
- Partner migration and FIPS compliance feature is modified to support OpenPGP.
- Partial document processing is not supported in OpenPGP.

There are some prerequisites that has to be performed before working with OpenPGP certificates.

Externally obtain the following library files, and copy them in HUB INSTALLED LOCATION>/lib/openpgp folder location:

- BouncyCastle OpenPGP library ver. 1.45 for JDK 1.5
- BouncyCastle JCE library ver. 1.45 for JDK 1.5

  **Important:** Obtain or procure these library files externally, as IBM does not ship them. For more information on obtaining the library files, see the home page link of Bouncy castle - http://www.bouncycastle.org. The jar files to extract are http://www.bouncycastle.org/download/bcpg-jdk15-145.jar and http://www.bouncycastle.org/download/bcprov-jdk15-145.jar. In case of distributed mode, place the jar files in all the computers where Document Manager and Console are installed.

After copying the files to the specified location, restart the server.

# Chapter 3. Creating and setting up partners

There are two types of partners: internal partners and external partners. The internal partner is typically the company that owns the WebSphere Partner Gateway server and that uses the server to communicate with other companies. The internal partner owns the backend applications (applications internal to the owning company). There can be any number of internal partners, but the default partner is typically the first partner defined. The other companies that the internal partner communicates with are the external partners.

For each partner with which you will be exchanging documents, you will need to create a partner profile. In addition to creating profiles, you will also need to set them up, a process that involves several required and optional steps.

This chapter outlines the basic steps of creating and setting up a partner profile. For more detailed information on a step, see the reference at the end of that step or section for more information. This chapter includes the following sections:



**Note:** You should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Creating partner profiles

### About this task

This is the first step in defining a partner in WebSphere Partner Gateway. This step defines basic information about the partner such as their name, login name, and business IDs.

To create a partner, you need to know the following information about the partner:

- The Business ID that the partner uses. This can be:
  - DUNS, which is the standard Dun & Bradstreet number associated with a company
  - DUNS+4, which is an extended version of the DUNS number
  - Freeform, which can be any number that the partner chooses to use to identify the company

For each partner that you want to add to the hub community, follow this procedure:

1. Click **Account Admin > Profiles > Partner**.
2. Click **Create**.
3. Enter **Company Login Name**. It is the name that the partner will use in the company field while logging into the hub. Blank spaces are not allowed in the company login name.
4. For **Partner Display Name**, enter the company name or some other descriptive name for the partner. This is the name that appears on the **Partner Search** list.
5. Select the type of partner. If this is the first partner, you will probably be setting up the company that owns WebSphere Partner Gateway. Therefore, you would choose **Internal Partner**. In the partner configuration screen, select the **Default Internal partner** check box if you want to set this current internal partner as default. When you select this check box for any other partner, the default selection is automatically removed from this internal partner. You cannot clear the selection in this page. For the first internal partner that is created, this checkbox is selected by default.
6. Optionally, enter the Admin User Name for the Administrator. The user Admin User Name is unique across all partners. The administrator for the partner can perform management activities for this partner, such as managing destinations, B2B capabilities, and users. The Hub Operator always has full access to partner management.
7. Select the status for the partner. Select **Enabled** if the status of the partner is Disabled. **Enabled** is the default status of the partner.
8. Optionally, enter the type of company in the **Vendor Type** field.
9. Optionally, enter the **Web Site** of the partner.
10. Click **Business ID** > **New**.
11. Specify a type from the list, and enter the appropriate identifier. WebSphere Partner Gateway uses the number you enter here to route the document to and from the partner.

    Observe the following guidelines when typing the identifier:

    a. DUNS numbers must equal nine digits.
    b. DUNS+4 must equal 13 digits.
    c. Freeform ID numbers accept up to 60 alphanumeric and special characters.

    **Note:** You can assign more than one business ID to a partner. In some cases, more than one Business ID is required. For example, when the hub sends and receives EDI X12 and EDIFACT documents, it uses both the DUNS and Freeform IDs during the document exchange.

    Both the internal partner and the external partners involved in these types of document flows should have both a DUNS and Freeform ID. The Freeform ID is used to represent EDI IDs that have both an identifier and a qualifier. For example, suppose the EDI qualifier is "ZZ" and the EDI identifier is "810810810". The Freeform ID could be specified as ZZ-810810810.
    When you click **New**, the Email ID text box also enables and displays for you to create an Email ID.
12. Click **New** to create a new Email ID and type your Email ID in Email Identifier. Similarly, you can click New to create multiple Email IDs.

13. Optionally, enter an IP address for the partner. The IP Address is used in conjunction with a destination when the "Validate Client IP" is configured. Enter an IP Address by performing the following steps:
    a. Under **IP Address**, click **New**.
    b. Specify the operation mode.
    c. Enter the IP address of the partner.
14. Click **Save**.
15. If you entered an Admin User Name, then you will be presented with a password that the partner will use to log on to the hub. Make a note of this password. You will provide it to the partner Admin user.

## Creating destinations

### About this task

After you create a profile for a partner, you need to establish the destinations that the hub will use to send documents to the partner.

Use the following procedure to create destinations for a partner:

1. Ensure that the partner profile for which you want to create destinations is selected.

   If you just created a profile, it is already selected. If it is not selected, follow these steps:
   a. Click **Account Admin** > **Profiles** > **Partner**.
   b. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
   c. Click the **View details** icon to display the partner's profile.
2. Click **Destinations**.
3. Click **Create**.
4. Type a **Destination Name** to identify the destination.
5. Optionally, indicate the **Status** of the destination.
6. Optionally, indicate whether the destination is **Online** or **Offline**.
7. Optionally, enter a **Description** of the destination.
8. Select a **Transport**.
9. After you select a transport, the **Destination Configuration** section of this page displays specific to that transport. For information on filling out this section for each transport, see one of these sections:
   - "Setting up global transport values" on page 210

     **Note:** These values pertain only to the FTP Scripting destination.
   - "Setting up an HTTP destination" on page 212
   - "Setting up an HTTPS destination" on page 213
   - "Setting up an FTP destination" on page 215
   - "Setting up an SMTP destination" on page 216
   - "Setting up a JMS destination" on page 217
   - "Setting up a file-directory destination" on page 220
   - "Setting up an FTPS destination" on page 221
   - "Setting up an FTP Scripting destination" on page 223
   - "Setting up SFTP destination" on page 222

# Setting up B2B capabilities

## About this task

Each partner has B2B capabilities that define the types of documents the partner can send and receive.

As the hub administrator, you can set up the B2B capabilities of your partners, or the partners can perform this task themselves. You use the B2B Capabilities feature to associate a partner's B2B capabilities with a document definition.

Use the following procedure to set the B2B capabilities of each partner:

1. Ensure that the partner profile for which you want to configure the B2B capabilities is selected. The selected profile is displayed at the top of the page after the **Profile**.

   If you just created a profile, it is already selected. If it is not selected, follow these steps to do so:

   a. Click **Account Admin**.
   b. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
   c. Click the **View details** icon to display the partner's profile.
2. Click **B2B Capabilities**. The B2B capabilities page is displayed. The right side of the page shows the packages, protocols, and documents supported by the system as document definitions.
3. Click the **Role is not active** icon under the **Set Source** column for the Packages. The package has documents that the external partners will send to the internal partner.
4. Select both **Set Source** and **Set Target** if the partners will send and receive those same documents. The Console displays a check if the document definition is enabled.

   **Note:** The selection of Set Source will be the same for all actions in 2-way PIP regardless of the fact that the request will originate from one partner and the corresponding confirmation from another. This also applies to Set Target.
5. Click the **Expand** icon at the **Package** level to expand an individual node to the appropriate document definition level or select a number from **0-4** or **All** to expand all displayed document definitions to the selected level.
6. Again, select the **Set Source**, **Set Target**, or both roles for the lower **Protocol** and **Document Type** levels for each document definition your system supports.

   If a definition is activated at the **Document Type** level, the **Action** and **Activity** definitions (if any exist) will be activated automatically.
7. Optionally, click **Enabled** under the **Enabled** column to place a document definition offline. (When you select **Set Source** or **Set Target**, the record is automatically enabled.) Click **Disabled** to place it online.

   If a package is disabled, all lower-level document definitions in that same node are also disabled, regardless of whether their individual status was enabled. If a lower-level document definition is disabled, all higher-level definitions within the same context remain enabled. When a document definition is disabled, all preexisting connections and attributes fails to function.
8. Optionally, click the **Edit** icon if you want to edit any of the attributes of a protocol, package, document type, action, activity, or signal. You then see the

settings for the attributes (if any attributes exist). You can modify the attributes by entering a value or selecting a value from the **Update** column and then clicking **Save**.

## Loading certificates

### About this task

Certificates enable partners to send and receive secure documents using several methods: encryption, digital signing, or SSL. Once a partner has received a certificate from another partner, that partner can use any of these methods to send the document.

Use the steps provided in "Uploading certificates using wizard" on page 265 to upload certificates for a partner.

For more information on using certificates, seeChapter 13, "Enabling security for document exchanges," on page 235.

## Creating users

### About this task

Users are the people who will log in to perform administration tasks for this partner. New users that are added to the LDAP server and WAS Admin console must also be added in the WebSphere Partner Gateway console in order to be active.

Use the following procedure to create users for a partner:

1. Ensure that the partner profile for which you want to create users is selected. The selected profile is displayed at the top of the page after **Profile >**. If the Profile name is not selected, follow these steps to create a profile:
   a. Click **Account Admin > Profiles > Partner**.
   b. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
   c. Click the **View details** icon to display the partner's profile.
2. Click **Users**.
3. Click **Create**.
4. Type the name of the user.

   **Note:** User names must be unique across all partners in the system.
5. Ensure that the status is **Enabled**.
6. Optionally, type the given name, family name, and other personal information of the user.
7. Select the **Language** and **Format Locales** and **Time Zone** of the user.
8. Change the Alert Status of the user status to **Enabled**.
9. Select the Subscribed Visibility of the user.
10. Either click **Auto Generate Password** to create a password for that user or type and retype one.
11. Click **Save**.

**Note:**

1. Because unique user names are required on an LDAP server, user names must be unique on WebSphere Partner Gateway as well. If you are creating a new user and the user name already exists in the same or for a different partner, you will see an error message stating that`A User with this name already exists`.
2. If you are migrating to WebSphere Partner Gateway from an earlier version where user names are not restricted, double asterisk (**) is displayed next to any duplicate user name showing that it also exists in the same or another partner profile. Change one of the user names so that they are unique from one another. New users and groups, which are added to the LDAP server and WAS Admin console, must also be added in the WebSphere Partner Gateway console in order to be active.

To enable LDAP to work with WebSphere Partner Gateway, you need to set up LDAP server authentication using the WebSphere Application Server console and LDAP user authorization using the WebSphere Partner Gateway Community Console. For information on setting up LDAP authentication, see the *WebSphere Partner Gateway Installation Guide*. For information on managing users and setting up LDAP user authorization, see the *WebSphere Partner Gateway Administration Guide*.

For more information on managing users, see "Managing users" in *WebSphere Partner Gateway Partner Guide*.

# FTP configuration

To configure an FTP or SFTP user, perform either of the following:

- “Creating FTP and SFTP users”Create a user in the FTP Management screen of the Console.
- “Enabling existing users for FTP and SFTP” on page 29

## Creating FTP and SFTP users

In this step, during creation, users are configured as FTP users or SFTP users.

### About this task

You can create FTP and SFTP users from the **FTP User Management** page of the console.

### Procedure

1. Click **Account Admin > FTP User Management**.
2. Click **Create**.
3. Enter the details of the user, and click **Save**. For more information about creating users, see “Creating users” on page 27. The information of the successfully created user is displayed in read-only mode.
4. Click **FTP Configuration** link.
5. In the FTP Configuration screen, select `Enabled` for **FTP user Enabled** or for **SFTP user Enabled**. You can enable a user for both FTP and SFTP Server.
6. Enter the following details of FTP configuration:
   a. Enter the **Home directory**, which is the relative path to the value specified for bcg.ftp.config.rootdirectory.
   b. Enable or disable **Write Permission** of the home directory.

c. Enable or disable the permission to **Create/Remove Directory**.
d. Select **Max Login Number**. It is the maximum number of times you can perform concurrent login.
e. Select **Max Login from Same IP**. It is the maximum number of times you can perform concurrent login from the same IP address.
f. Select **Max Idle Time (Seconds)**. It is the maximum idle time in seconds after which the user connection is discarded.
g. Select **Max. upload (bytes/sec)**. It is the maximum rate of upload in bytes/sec.
h. Select **Max. Download (bytes/sec)**. It is the maximum rate of download in bytes/sec.

**Note:** Some fields have `Custom Limit` value in the drop down list. If you select `Custom Limit` from the drop down list, then enter the customized value in the text box.

7. For SFTP configuration, enter **Key (SFTP only)**. The uploaded file is used for key based authentication. The folder icon indicates that a Key is already uploaded. You can also use **Browse** to upload a key.
8. Click **Save**.

## Enabling existing users for FTP and SFTP

In this step, you can set an existing user as an FTP user or an SFTP user.

### About this task

To configure an FTP or SFTP user, enable FTP or SFTP properties for an existing user.

### Procedure

1. Click **Account Admin > Profiles > Users**.
2. Enter the search criteria and click **Search**.
3. In the search results, if the **Status** column is disabled for the contact, click the `Enabled icon`. The icon toggles between enable and disable states.
4. Click **View Details icon** for the user to configure FTP access.
5. In the user details screen, click **FTP Configuration link.**
6. In the **FTP Configuration** screen, select `Enabled` for **FTP user Enabled** or for **SFTP user Enabled**. A user can be enabled for both FTP and SFTP Server.
7. Enter the details of FTP or SFTP configuration. See “Creating FTP and SFTP users” on page 28 for the details of FTP and SFTP user.
8. Click **Save**.

# Creating groups

### About this task

Grouping users enables you to manage the permissions of many users at once. New groups that are added to the LDAP server and WebSphere Application Server Admin console must also be added in the WebSphere Partner Gateway console in order to be active.

Use the following procedure to create groups for each partner:

1. Ensure that the partner profile for which you want to create groups is selected.

   If you just created a profile, it is already selected. If it is not selected, follow these steps to do so:

   a. Click **Account Admin** > **Profiles** > **Partner**.
   b. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
   c. Click the **View details** icon to display the partner's profile.
2. Click **Groups**.
3. Click **Create**.
4. Type the name of this group.
5. Click **Save**.
6. To add users to this group, click **Memberships** link.

   Users that are associated with this partner are displayed under **Users Not in Group** or **Users in Group**. To add a user to a group, complete the following,

   a. Click the **Edit Record** icon beside the group.
   b. Select the user you want to add, and click **Add to Group**.
   c. Click **Save**.
7. To change the permissions of the users in this group, click **Permissions** link.

   Permissions for the users of this group are displayed by **Module**. To change the permissions of this group, complete the following,

   a. Click the **Edit Record** icon beside the group.
   b. Click the radio buttons to the right of each module specifying the permission as **No Access**, **Read Only**, or **Read/Write**.
   c. Click **Save**.

   **Note:** Users can belong to more than one group. In these cases when the permissions in the different groups differ, the user inherits the highest level of permissions assigned to the users in all groups.

**Note:** All the members of the hubadmin group can have superuser permissions. This allows for many people to share hubadmin responsibilities while maintaining password security.

For more information on managing groups, see "Managing groups" in *WebSphere Partner Gateway Partner Guide*.

## Creating contacts

### About this task

WebSphere Partner Gateway allows you to create contacts who can be notified when different types of events occur. Use the following procedure to create contacts for each partner:

1. Ensure that the partner profile for which you want to create contacts is selected. The selected profile is displayed at the top of the page after **Profile >**.

   If the profile is not selected, perform the following steps:

   a. Click **Account Admin > Profiles > Partner**.
   b. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.

c. Click the **View details** icon to display the partner's profile.

2. Click **Contacts**.
3. Click **Create**.
4. Type the **Given Name** and **Family Name** of this contact.
5. Optionally, type the **Address** of this contact.
6. Optionally, select the **Contact Type**.
7. Optionally, type the **E-mail** address, **Telephone** number, and **Fax Number** of this contact.
8. Select the **Language** and **Format Locales** and **Time Zone** of the contact.
9. Change the **Alert Status** of the user status to **Enabled**.
10. Select the **Subscribed Visibility** of the user.
11. Click **Save**.

For more information on managing contacts, see "Managing contacts" in *WebSphere Partner Gateway Partner Guide*.

# Creating addresses

## About this task

WebSphere Partner Gateway allows to create addresses for partners. Use the following procedure to create an address for a partner:

1. Ensure that the partner profile for which you want to create addresses is selected. The selected profile is displayed at the top of the page after **Profile >**.

   If you just created a profile, it is already selected. If it is not selected, follow these steps to do so:

   a. Click **Account Admin > Profiles > Partner**.
   b. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
   c. Click the **View details** icon to display the partner's profile.
2. Click **Addresses**.
3. Click **Create New Address**.
4. Select an **Address Type**.
5. Optionally, type the **Address**.
6. Click **Save**.

For more information on managing addresses, see "Managing addresses" in *WebSphere Partner Gateway Partner Guide*.

# Chapter 4. Preparing to configure the hub

In the next few chapters, you will be setting up the receivers and destinations described in Chapter 2, "Introduction to hub configuration," on page 5. Depending on the transport types used to send and receive documents, you need to setup receivers and destinations.

This chapter covers the following topics:

- " Creating a file-directory destination"
- "Configuring the FTP server for receiving documents"
- "Configuring the hub for the JMS transport protocol" on page 37
- "Configuring RNIF compression" on page 42

It also provides a brief overview of the FTP scripts needed for the FTP Scripting receivers and destinations, and it describes the Data Interchange Services client, which can be used to create transformation, validation, and functional acknowledgment maps for EDI, XML, and record-oriented-data (ROD) documents.

- "Using FTP scripts for FTP Scripting receivers and destinations" on page 43
- "Using maps from the Data Interchange Services client" on page 43

If you are not planning to set up any of these types of receivers or destinations, skip this chapter and go to Chapter 5, "Starting the server and displaying the Community Console," on page 45.

## Creating a file-directory destination

The directory that you specify for a file-directory destination will be created for you if necessary. If it already exists, then it will be used by the destination.

## Configuring the FTP server for receiving documents

**Note:** This section applies only to receiving documents over FTP or FTPS from partners. Sending documents to partners is described in "Setting up an FTP destination" on page 215 and "Setting up an FTPS destination" on page 221.

If you are going to use FTP or FTPS as a transport for incoming documents, you must have an FTP server installed. If you are planning to use FTP and do not currently have a server installed, do so now before continuing. Make sure that one of the following scenarios is true for your installation:

- The FTP server is installed on the same machine on which WebSphere Partner Gateway is installed.
- The bcguser on the WebSphere Partner Gateway machine has read/write access to the location where the FTP server will be storing files.

**Note:** If the installation setup is on multiple machines, FTP server has to be installed where the receiver is installed.

## Configuring the required directory structure on the FTP server

### About this task

After the FTP server is installed, the next step is to create the required directory structure under the home directory of the FTP server. WebSphere Partner Gateway requires a particular directory structure that the Receiver component and Document Manager components use to correctly identify the partner sending the incoming document. The structure is illustrated in Figure 15.

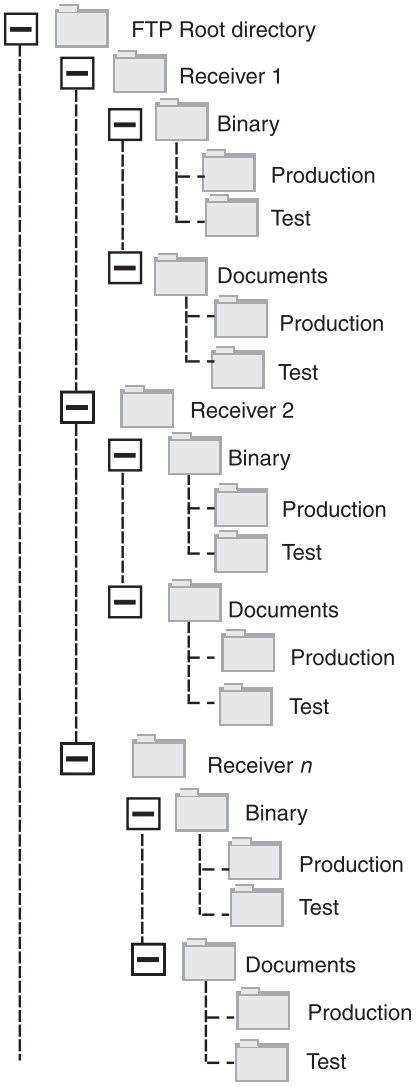



*Figure 15. FTP directory structure*

Each partner directory contains a Binary directory and a Documents directory. Both the Binary and Documents directories contain a Production directory and a Test directory.

The Documents directory is used when a partner sends an XML document containing complete routing information (using FTP) to the hub. This requires the creation of a custom XML definition. Also, Electronic Data Interchange (EDI) documents can be sent using this directory.

The Binary directory is used when a partner sends any other documents (using FTP) to the hub.

For each partner who will use FTP to send or receive documents, create the following folders from the root directory of your FTP server:

1. Create a folder for the partner.

   **Note:** The name of the folder should match the name you specify for **Company Login Name** when you create the partner. Creating partners is described in "Creating partner profiles" on page 23.
2. Create subfolders under the partner folder named `Binary` and `Documents`.
3. Create subfolders under the Binary and Documents folders called `Production` and `Test`.

## Processing files that are sent over FTP

It is important to understand how binary and XML files are processed by the FTP server.

### Binary files

Binary files have a required file name structure, because the files are not inspected at all by the Document Manager.

The file name structure is: *<To_PartnerID>.<Unique_Filename>*

When a binary file is detected by the Receiver component, it is written to shared storage and passed to the Document Manager for processing.

The name of the directory in which the file was detected is used to evaluate the From Partner Name, and the first part of the file name is used to evaluate the To Partner Name. The position of the directory in the directory structure is used to evaluate whether the transaction is a Production or Test transaction.

For example, a file named 123456789.abcdefg1234567 is detected in the `\ftproot\partnerTwo\binary\production` directory. The Document Manager knows the following information:

- The `From Partner Name` is `partnerTwo` (because the file was found in the partnerTwo part of the directory tree).
- The `To Partner Name` is `partnerOne` (because the first part of the file name is 123456789, which is the DUNS ID for partnerOne).

  **Note:** Here and throughout this book, all DUNS numbers are meant to be examples only. WebSphere Partner Gateway requires the <To_PartnerID> to match the recipient partner DUNS. In case the Duns ID is not found, the channel lookup will fail.
- The Transaction type is Production.

The Document Manager looks for a Production partner connection from partnerTwo to partnerOne for:

- Package: None (N/A)
- Protocol: Binary (1.0)
- Document Type: Binary (1.0)

The Document Manager then processes the file.

Binary files can also be transferred over FTP using either the Generic Preprocess handler or FileNamePartnerId handler. See “Modifying the preprocess configuration point” on page 74 for more details.

### XML files

An XML file that is routed using your custom XML specifications has no file name requirements because the file is inspected by the Document Manager and the routing information is extracted from the document itself.

When a XML file is detected by the Receiver component, it is written to the shared storage and passed to the Document Manager for processing.

The Document Manager compares the XML file to the XML Formats that have been defined and selects the required XML Format. (Setting up XML formats is described in “Custom XML document processing” on page 148.) The From Partner Name, To Partner Name, and the Routing information are extracted from the XML File.

The position of the directory in the directory structure is used to evaluate whether the transaction is a Production or Test transaction.

The Document Manager then uses this information to locate the correct partner connection before processing the file.

## Additional FTP server configuration

### About this task

After creating the required directory structure, you configure your FTP server for each of the partners in the hub community. The way you configure the FTP server depends on which server you are using. Refer to the FTP server documentation, and perform the following tasks:

### Procedure

1. Add a new group (for example, Partners).
2. Add a user to the newly created group for each partner who will be sending or receiving documents over FTP.
3. For each partner, set up the FTP server to map the incoming partner to the directory structure that you have created for the partner in the earlier section “ Configuring the required directory structure on the FTP server” on page 34. Refer to your FTP server documentation for additional information.

## Security considerations for the FTPS server

If you are using an FTPS server to receive incoming documents, the security considerations for the SSL sessions are handled solely by the FTPS server and client that the partner is using. There is no specific security configuration for WebSphere Partner Gateway on incoming FTPS documents. WebSphere Partner Gateway retrieves the documents from the FTP receiver (which is described in “ Setting up an FTP receiver” on page 59) after the server has successfully negotiated the secure channels and received the document. Refer to the FTPS server documentation to determine which certificates are needed (and where they are needed) to successfully configure a secure channel that the partner can contact.

For server authentication, provide the certificate of the Receiver component to the partners. If the certificate is issued by a Certifying Authority (CA), also provide the

CA certificate chain. If client authentication is supported by the FTPS server, the client authentication certificates of the partners should be specified in the FTPS server. Consult the FTPS server documentation for information about specifying client authentication and client authentication certificates.

# Configuring the hub for the JMS transport protocol

This section describes how to set up the hub to use the JMS transport. If you will be using the JMS transport to send documents from the hub or to receive documents at the hub, follow the procedures in this section. If you will not be using the JMS transport, skip this section.

**Note:** The procedures in this section describe how to use the JMS implementation of WebSphere MQ to set up the JMS environment. The procedures also describe how to set up local queues. If you want to set up transmission and remote queues, refer to the WebSphere MQ documentation.

Even though this section is specific to WebSphere MQ, other JMS providers will require similar procedures. For WebSphere Platform Messaging, see "Configuring JMS while WebSphere Partner Gateway is installed on WebSphere Application Server" in Chapter 5. "Integrating WebSphere Process Server with JMS as transport" in the *WebSphere Partner Gateway Integration Guide*.

In later sections of this document, you will learn how to set up JMS receivers or destinations (or both). These tasks are described in " Setting up a JMS receiver" on page 62 and "Setting up a JMS destination" on page 217.

## Creating a directory for JMS

### About this task

You first create a directory for JMS. For example, suppose you wanted to create a directory named JMS in the c:\temp directory of a Windows installation. These are the steps you would follow:

### Procedure

1. Open Windows Explorer.
2. Open the C:\temp directory.
3. Create a new folder named JMS.

## Modifying the default JMS configuration

### About this task

In this section, you update the JMSAdmin.config file, which is part of the WebSphere MQ installation, to change the context factory and provider URL.

1. Navigate to the `Java\bin` directory of WebSphere MQ. For example, in a Windows installation, you would navigate to: C:\IBM\MQ\Java\bin
2. Open the JMSAdmin.config file using a plain text editor, such as Notepad or vi.
3. Add the character # to the front of the following lines:

   ```
   INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
   PROVIDER_URL=ldap://polaris/o=ibm,c=us
   ```

4. Remove the character # from the front of the following lines:

   ```
   #INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory
   #PROVIDER_URL=file:/C:/JNDI-Directory
   ```

5. Change the `PROVIDER_URL=file:/C:/JNDI-Directory` line to equal the name of the JMS directory you set up in " Creating a directory for JMS" on page 37. For example, if you set up the `c:/temp/JMS` directory, the line would look like this:

   ```
   PROVIDER_URL=file:/c:/temp/JMS
   ```

6. Save the file.

## Creating queues and the channel

In this section, you use WebSphere MQ to create the queues you will use to send and receive documents and the channel for this communication. It is assumed that a queue manager has been created. The name of the queue manager should be substituted where *<queue_manager_name>* appears in the following steps. It is also assumed that a listener for this queue manager has been started on TCP port 1414.

1. Open a command prompt.
2. Enter the following command to start the WebSphere MQ command server:

   ```
   strmqcsv <queue_manager_name>
   ```

3. Enter the following command to start the WebSphere MQ command environment:

   ```
   runmqsc <queue_manager_name>
   ```

4. Enter the following command to create a WebSphere MQ queue to be used to hold incoming documents sent to the hub:

   ```
   def ql(<queue_name>)
   ```

   For example, to create a queue named JMSIN, you would enter:

   ```
   def ql(JMSIN)
   ```

5. Enter the following command to create a WebSphere MQ queue to be used to hold documents sent from the hub:

   ```
   def ql(<queue_name>)
   ```

   For example, to create a queue named JMSOUT, you would enter:

   ```
   def ql(JMSOUT)
   ```

6. Enter the following command to create a WebSphere MQ channel to be used for documents sent to and from the hub:

   ```
   def channel(<channel_name>) CHLTYPE(SVRCONN)
   ```

   For example, to create a channel named java.channel, you would enter:

   ```
   def channel(java.channel) CHLTYPE(SVRCONN)
   ```

7. Enter the following command to exit the WebSphere MQ command environment:

   ```
   end
   ```

## Adding a Java run time to your environment

### About this task

Enter the following command to add a Java(TM) run time to your system path:

```
set PATH=<ProductDir>\_jvm\jre\bin
```

where *ProductDir* refers to the directory where WebSphere® Partner Gateway is installed.

## Defining the JMS configuration

### About this task

To define the JMS configuration, perform the following steps:

1. Change to the WebSphere MQ Java directory (directory (<*path_to_Websphere_MQ_installation_directory*>\java\bin)
2. Start the JMSAdmin application by typing the following command:

   ```
   JMSAdmin
   ```

3. Define a new JMS Context by typing the following commands from the InitCtx> prompt:

   ```
   define ctx(<context_name>)
   change ctx(<context_name>)
   ```

   For example, if the *context_name* is JMS, the commands look like this:

   ```
   define ctx(JMS)
   change ctx(JMS)
   ```

4. From the InitCtx/jms> prompt, enter the following JMS configuration:

   ```
   define qcf(connection_factory_name)
       tran(CLIENT)
       host(<your_IP_address>)
       port(1414)
       chan(java.channel)
       qmgr(<queue_manager_name>)
   define q(<name>) queue(<queue_name>) qmgr(<queue_manager_name>)
   define q(<name>) queue(<queue_name>) qmgr(<queue_manager_name>)
   end
   ```

   **Note:**

   - If MQ and WebSphere Partner Gateway are installed on two different machines, select the Transport type as CLIENT.
   - If MQ and WebSphere Partner Gateway are installed on the same machine, then the Transport type must be BINDINGS.

The previous steps created the .bindings file, which is located in a subfolder of the folder you specified in step 5 on page 38. The name of the subfolder is the name you specified for your JMS context.

As an example, the following JMSAdmin session is used to define the queue connection factory as `Hub`, with an IP address of `sample.ibm.com` where the MQ queue manager resides (<*queue_manager_name*> of `sample.queue.manager`). The example uses the JMS queue names and channel name created in "Creating queues and the channel" on page 38. Note that user input follows the > prompt.

```
InitCtx> define ctx(jms)
InitCtx> change ctx(jms)
InitCtx/jms> define qcf(Hub)
    tran(CLIENT)
    host(sample.ibm.com)
    port(1414)
    chan(java.channel)
    qmgr(sample.queue.manager)
InitCtx/jms> define q(inQ) queue(JMSIN) qmgr(sample.queue.manager)
InitCtx/jms> define q(outQ) queue(JMSOUT) qmgr(sample.queue.manager)
InitCtx/jms>end
```

In this example, the .bindings file would be located in the following directory: `c:/temp/JMS/JMS`, where `c:/temp/JMS` is the PROVIDER_URL and `JMS` is the context name.

## Configuring the runtime libraries

For the JMS Receiver or JMS Destination, there are several WebSphere MQ jar files that need to be visible to WebSphere Partner Gateway. These jar files are made visible by putting them into the class path. If you will be using MQ Binding mode to access MQ, then the MQ native libraries are also required to be in the path. Refer to the WebSphere MQ documentation for further information on the MQ jar files and native libraries for JMS.

There are several ways to add the jar files to Websphere Partner Gateway class path. One way is to put them into the User Exits directory and the second is to associate them via a WebSphere Application Server shared libraries.

### User Exits directory method

To use this method, put the specified jar files in the appropriate User Exits directory:

- For the JMS Receiver put them into the <WPG-Install root>/receiver/lib/userexits directory
- For the JMS Destination put them into the <WPG-Install root>/router/lib/userexits director

### WebSphere Application Server shared libraries method

#### About this task

To use this method, create a shared library variable, and then associate the variable with the Receiver or Document Manager Application as shown briefly in the following steps. Refer to the WebSphere Application Server documentation for more information on this procedure.

1. Log into the WebSphere Application Server administrative console.
2. Create the Shared Libraries variable by completing the following:
   a. Navigate to **Environment** > **Shared Libraries**.
   b. Select a **Scope** (probably node), and click **New**.
   c. Enter the name of the variable (for example, MQ_LIBRARIES), complete the class path entries for the MQ jar files, and click **OK**.
3. Associate the shared library variable that you created with the WebSphere Partner Gateway components by completing the following:
   a. Navigate to **Applications** > **Enterprise Applications**.
   b. Select either **BCGReceiver** (for JMS Receivers) or **BCGDocMgr** (for JMS Destinations).
   c. Select **Shared Library References**.
   d. Select the application, and click **Reference Shared Libraries**.
   e. From the Available list, select the shared library variable that you created (for example, MQ_LIBRARIES), and move the variable to the Selected list. Then click **OK**.

### Configuring JMS gateway and receiver with external MQ

#### About this task

Following are the steps to create a communication bridge between WebSphere Partner Gateway and MQ through WebSphere Application Server administrative console:

1. Create JMS Queue Connection Factory.
   a. Log into the WebSphere Application Server administrative console.

b. Navigate to **Resources > JMS > Queue connection factories**.
c. Select a **Scope** and click **New.**
   - For gateway configuration, select the scope of document manager server/node. (Node scope is helpful in case of clusters. For simple mode, select a server scope.)
   - For receiver configuration, select the scope of receiver server/node. (Node scope is helpful in case of clusters. For simple mode, select a server scope.)
d. Select the option **WebSphere MQ messaging provider** and click **OK**.
e. Enter the **Name** and **JNDI Name**. These are required values.
f. Enter appropriate values for **Queue Manager**, **Host** (IP of the machine where queue manager is running), **Port**, **Channel**, and **Transport type**. Rest of the fields are optional.

   **Note:**
   - If MQ and WebSphere Partner Gateway are installed on two different machines, select the Transport type as CLIENT.
   - If MQ and WebSphere Partner Gateway are installed on the same machine, then the Transport type must be BINDINGS.

   For more details, refer to the WebSphere Application Server InfoCenter at the following site: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.multipla...

2. Create JMS Queue.
   a. Log into WebSphere Application Server administrative console.
   b. Navigate to **Resources > JMS > Queues**.
   c. Select a **Scope** and click **New.**
      - For gateway configuration, select the scope of document manager server/node. (Node scope is helpful in case of clusters. For simple mode, select a server scope.)
      - For receiver configuration, select the scope of receiver server/node. (Node scope is helpful in case of clusters. For simple mode, select a server scope.)
   d. Enter the **Name** and **JNDI Name**. These are required values.
   e. Enter appropriate values for **Queue Manager**, **Host** (IP of the machine where queue manager is running), **Port**, **Channel**, and **Transport type**. Rest of the fields are optional.
   f. Restart the servers that have undergone change, for example DocumentManager/Receiver/bcgserver in case of simple distributed installation.
3. Configure JMS Gateway on WebSphere Partner Gateway.
   a. Log into WebSphere Partner Gateway administrative console.
   b. Click **Account Admin > Profiles > Destinations**.
   c. Click **Create**.
   d. Enter the **Destination name**. This is a required field.
   e. Select **JMS** in the transport field.
   f. Enter values for the following mandatory fields:
      - Address: Enter the destination address by providing the appropriate hostname and port of the Queue Connection Factory or Queue objects,

which were created in WebSphere Application Server. The address must be in the format corbaloc:iiop: <hostname>: <bootstrapportnumber>, where:

- corbaloc:iiop - signify the protocol used for communication between Client (WebSphere Partner Gateway) and Server look-up (WebSphere Application Server).
- <hostname> - hostname or IP address of the machine where WebSphere Application Server is installed, for which Queue Connection factory and Queue objects were created.
- <bootstrapportnumber> - the bootstrap port number of the Server where Queue Connection Factory and Queue objects are bound together. To get the bootstrap port number, you can log into WebSphere Application Server administrative console, navigate to **Servers > Application Server > <server name> Ports** and check the bootstrap address. In case of distributed mode, port numbers are different for Receiver and Gateway. Access the corresponding server (bcgreceiver for Receiver and bcgdocmgr for Gateway) to get the correct bootstrap port number.

- JMS Factory Name: the JNDI name provided for JMS Queue Connection Factory.
- JMS Queue Name: the JNDI name provided for JMS Queue.
- JMS JNDI Factory Name: is the factory to be used for JNDI communication. Since you are using WebSphere Application Server, you can specify the value as com.ibm.websphere.naming.WsnInitialContextFactory.

4. Configure JMS Receiver on WebSphere Partner Gateway.
   a. Log into WebSphere Partner Gateway administrative console.
   b. Click **Hub Admin > Hub Configuration > Receivers**.
   c. Click **Create Receiver**.
   d. Enter the **Receiver Name**. This is a required field.
   e. Select **JMS** in the transport field.
   f. Enter appropriate values for the required fields as described in the step :3f on page 41.

## Configuring RNIF compression

Rosettanet Business messages and their attachments are compressed and packaged using S/MIME envelope to transfer large documents. Also, the decompression support is provided for Rosettanet Business messages. You are provided with an option to compress the payload either alone or along with attachments. For performance improvement, compress the service content and its attachments prior to encryption, signing, or transfer encoding as per Rosettanet 2.0 Technical Advisory Specification. Under the appropriate Rosettanet WebSphere Partner Gateway channel, select the routing object attribute compression to have any one of the following values:

- None
- Payload
- Payload and attachment

Apart from the selected compression option, you can also select additional filter criteria attributes such as **Compress Content Type** and **Compress Size**. You can select the payload or attachments for compression from the pool of attachments by

using the filter criteria. The **Compress Content Type** expects either "All" or valid Mime Types separated by commas. If you select **Payload** option in your base compression, then payload would be compressed irrespective of the value specified in **Compress Content Type** routing object attribute. Only attachments are selected for compression based on specified content types. Routing object attribute **Compress size** expects either "All" or valid size limit. The valid size limit denotes the minimum size acceptable for compression.

When a compressed Rosettanet document is sent, S/MIME decompression is performed over service content and its attachments.

## Using FTP scripts for FTP Scripting receivers and destinations

The FTP Scripting transport allows you to send data to any FTP service, including a Value Added Network (VAN). You control the operations on the FTP server using a script file that contains FTP commands.

You specify this script when you create the FTP Scripting receiver or destination. WebSphere Partner Gateway substitutes the actual values you enter when you create the receiver or destination for the placeholders in the FTP script.

The operations defined in the input script are translated into actions on the FTP server. The input script is made up of a group of supported FTP commands. Parameters for these commands can take the form of a variable, which is filled in at runtime.

For information about creating an FTP script for an FTP Scripting receiver, see " Setting up an FTP Scripting receiver" on page 65. For information about creating an FTP script for an FTP Scripting destination, see "Setting up an FTP Scripting destination" on page 223.

## Using maps from the Data Interchange Services client

To perform EDI de-enveloping, transformation, and validation or to make transformations between ROD, XML, and EDI, you need to import the associated maps from the Data Interchange Services client. Data Interchange Services is a separately installed program that typically resides on a different computer from the one on which WebSphere Partner Gateway runs.

The Data Interchange Services mapping specialist creates maps describing how specific documents should be transformed and validated.

To create any map, the definition of source and target document are required. The definitions of source documents, for EDI is supplied by WDI, whereas for ROD and XML, you need to create using DIS client. For EDI, import the .eif file, standard file, into DIS client. In case of ROD, create the standard using DIS client. Import the DTD/XSD to create standard for XML. The standard and transformation map can be compiled separately.

For example, you might have a purchase order created by a back-end application that you want transformed and sent to an external partner as a standard EDI X12 purchase order (850). The Data Interchange Services mapping specialist would write a map detailing how to transform each field or piece of data from your program to the X12 format. The map would then be exported directly to WebSphere Partner Gateway, or it would be exported to a file, which you would then import using a command script.

Detailed information about how to import maps from the Data Interchange Services client is provided in “ Importing maps manually” on page 194.

**Note:** The DIS client has its own database. After you complete a map in DIS client, export it as an .EIF file. From the console of WebSphere Partner Gateway, import this . EIF file. It will store the information in the WebSphere Partner Gateway database.

## Completing post-installation configuration tasks

After you have installed WebSphere Partner Gateway, you need to configure it. Typically this configuration involves using the WebSphere Partner Gateway administrative console to set up your hub. Depending on your trading community requirements, you may also need to configure the WebSphere Application Server infrastructure that hosts the WebSphere Partner Gateway components. Several such tasks are listed here along with links to detailed instructions for performing each task.

- “Changing the cryptographic strength” on page 244
- “SSL with Client Authentication configuration” on page 246

# Chapter 5. Starting the server and displaying the Community Console

This chapter shows you how to start the WebSphere Partner Gateway server and display the Community Console. It includes the following topics:

- "Starting the WebSphere Partner Gateway components"
- " Logging in to the Community Console" on page 47

For information on how to start the Clusters from the WebSphere Application Server Network Deployment administrative console, see Chapter 1. "Managing the WebSphere Partner Gateway component applications" of the *WebSphere Partner Gateway Administration Guide*.

## Starting the WebSphere Partner Gateway components

### About this task

To start the server, you must start each of the three components of WebSphere Partner Gateway: the Console, the Document Manager, and the Receiver.

In simple mode, all of the WebSphere Partner Gateway components are installed on the same instance of WebSphere Application Server. You start and stop all of the components using scripts and WebSphere Application Server administrative console. To start the WebSphere Partner Gateway components in a simple mode system, run the script:

```
<INSTALL DIR>/bin/bcgStartServer.sh
```

To stop the WebSphere Partner Gateway components in a simple mode system, run the script:

```
<INSTALL DIR>/bin/bcgStopServer.sh
```

**Note:** You do not have to specify a server name while installing in simple mode. When you install in simple mode, the server name is always server1.

**Note:** If the installer is run when the **temp** directory is low on space and the installer fails to install the product correctly, increase the space in the **temp** directory, and uninstall and reinstall the product.

1. Type http://<computer name or IP address>:58080/console, the Web browser displays the Welcome page. Log into WebSphere Partner Gateway using the following information:
   - In the **User Name** field, type:
     `hubadmin`
   - In the **Password** field, type:
     `Pa55word`
   - In the **Company Login Name** field, type:
     `Operator`

   Click **Login**.
2. When you log in for the first time you must create a new password. Type a new password, then type the new password a second time in the **Verify** field.

3. Click **Save**. The system displays the Community Console's initial entry window.

When you install WebSphere Partner Gateway Application, the First steps application and Installation Verification Test (IVT) application are installed by default. It will remain installed until the very last WebSphere Partner Gateway component resides in the machine. First steps page populates the data of installed components to run the verification test for each component separately.

You can invoke First steps page using the command **bcgFirstSteps. sh**, available in <install_dir>/FirstSteps/bin folder.

From the First steps page of the console, start and stop options can be toggled for all the installed components. For example, if the hub is running, the stop option will be listed. Otherwise, the start option is listed. The following lists the start and stop options of the components based on their topologies:

- The start and stop for Web Sphere Process Gateway is available for simple and simple distributed topologies.
- The start and stop for MAS is available for simple distributed and fully distributed topologies.
- The start and stop for Deployment Manager is available for simple and fully distributed topology.

  **Note:** This option will be available only if the Deployment Manager is installed using WebSphere Partner Gateway installer.
- The start and stop of console, Receiver and router are available for Fully distributed topology.
- The start and stop of FTP Management is available for all topologies.

The above options are available for the listed topologies provided they are installed in that machine. You have to verify the server logs for the successful completion of the action. You can also refer to the command line window to check the status. When you click **Start WPG** link in the First Steps panel, the start command is issued in DOS command prompt. The first steps panel will not notify you about the successful (or unsuccessfully) completion of the command.

When the install verification test (IVT) option is invoked, it verifies the validity of WebSphere Partner Gateway components that are installed in the machine. This verification test can be alternatively invoked from command line using **LaunchIVT.sh** command. This command is present in <installdir>/FirstSteps/ivt/bin folder. After the completion of verification, the IVT produces a report containing the details of all the installed WebSphere Partner Gateway components. Also, it cleans the temporary files that were created during this operation and stops any servers/nodes that were started for this operation. To indicate failure of any component, necessary log files are produced in <installdir>/FirstSteps/ivt/logs folder.

**Note:** In distributed topology, IVT will not verify components installed in different machines.

When you are trying to upload certificates with stronger cryptography than the default cryptography, the certificates can fail to upload.

# Logging in to the Community Console

## About this task

This section provides the steps for displaying and logging into the Community Console. The recommended screen resolution is 1024x768.

**Note:** The WebSphere Partner Gateway Community Console requires cookie support to be turned on to maintain session information. No personal information is stored in the cookie and it expires when the browser is closed.

1. Open a Web browser and enter the following URL to display the console:

   http://<*hostname*>.<*domain*>:58080/console (unsecure)

   https://<*hostname*>.<*domain*>:58443/console (secure)

   Where <*hostname*> and <*domain*> are the name and location of the computer hosting the Community Console component.

   **Note:** These URLs assume the default port numbers are used. If you changed the default port numbers, replace the default numbers with the values you specified.

In most cases, your hub administrator has sent you the user name, initial password, and company login name that you will use to log in to the Community Console. You will need this information for the following procedure. If you have not received this information, contact your hub administrator.

To log in to the Community Console (these instructions are for the internal partners as well as external partners):

1. Enter the **User Name** for your company.
2. Enter the **Password** for your company.
3. Enter your **Company Login Name**, for example, IBM.
4. Click **Login**. When you log in the first time, you must create a new password.
5. Enter a new password, then enter the new password a second time in the Verify text box.
6. Click **Save**. The system displays the console's initial entry screen.

**Note:** If WebSphere Partner Gateway is configured using LDAP, then you have to enter the LDAP User Name and Password. The Company Login Name is not relevant in this scenario, hence you will not be prompted to enter this information. Also, the system will not prompt you to change your password.

# Chapter 6. Configuring the Community Console

This chapter describes how to configure the Community Console to specify what partners see, how they log in to the console, and what access they have to various console tasks. This chapter includes the following topics:

- "Specifying locale information and console branding"
- "Setting the password policy" on page 51
- "Configuring permissions" on page 52
- "Setting console time out value" on page 53

You do not have to perform any of these tasks if you want to use the default settings supplied by WebSphere Partner Gateway.

**Note:** You should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Specifying locale information and console branding

### About this task

By default, the pages of the Community Console are presented in the English language. IBM provides translations of the content in other languages as a set of files that can be uploaded. Other console items that are provided by IBM for different locales are the banner graphics. Optionally, you can upload your own logo graphics. You can also upload your own custom style sheet used to format the text on the pages.

You perform these tasks using the Locale Upload page. To display the Locale Upload page:

1. Click **Hub Admin > Console Configuration > Locale Configuration**.
2. Click **Create**.
3. Select a locale from the **Locale** list.

The Console displays the Locale Upload page.

From the Locale Upload page, you can choose to perform the following tasks:

- Brand the console, by uploading a unique banner or logo (or both).
- Upload files that IBM provides so that you can localize the content of the elements on the console.

## Branding the console

### About this task

You can customize the way the Community Console looks by changing the branding images. Branding of the Community Console consists of importing two images: header background and company logo.

- The header background spans the top of the Community Console.

- The company logo is displayed at the top right of the Community Console.

The images must be .JPG format files and must conform to certain specifications, to fit in the Community Console window.

- To see the specifications required for the banner and logo, click **Image Specifications** on the Locale Upload window.
- To see samples of a header or logo image, scroll down to the **Sample Images** portion of the page and click **sample_headerback.jpg** or **sample_logo.jpg**.
- To download samples of a banner and logo to use as a template for creating your own banner and logo, click **Sample images (header background and company logo)**.

After you have created the banner or logo (or both), perform the following steps:

1. To upload the customized banner, perform either of the following tasks:
   - In the **Banner** field, type the path and name of the image file you want to use for the header/banner.
   - Click **Browse** to navigate to the .jpg file containing the banner, and select it.
2. To upload the customized logo, perform either of the following steps:
   - In the **Logo** field, type the path and name of the file you want to use for the company logo.
   - Click **Browse** to navigate to the .jpg file containing the logo, and select it.
3. Click **Upload**.

**Note:** When you replace the header background and company logo, you must restart the Community Console for the changes to take effect.

## Changing the style sheet

### About this task

If you want to specify a style sheet that is different from the default (for example, if you want different sized fonts or colors), perform the following tasks:

1. Perform one of the following tasks:
   - In the **CSS** field, type the path and name of the file that contains the customized style sheet.
   - Click **Browse** to navigate to the file containing the style sheet, and select it.
2. Click **Upload**.

## Localizing the data on the console

### About this task

If you receive resource bundles or other locale files from IBM, you can use the Locale Upload page to upload them. Resource bundles include the following information:

- **Console Labels**, which contain text strings that represent all the text on the interface
- **Event Descriptions**, which contain text strings used to display event details (for example, “An attempt was made to create a duplicate connection”)
- **Event Names**, which contain text strings representing event names (for example, “Connection already exists”)

- **EDI Event Descriptions**, which contain text strings used to display EDI event details (for example, “FA Reconciliation Failure. No activity ids found for the transactions found in the EDI Acknowledgement. “)
- **EDI Event Names**, which contain text strings representing EDI event names (such as “FA Reconciliation Failure”)
- **Extended Event Text**, which contain text strings that provide additional information about events (for example, the cause of the event and troubleshooting information)

To upload a resource bundle or other locale file:

1. For each resource bundle or file, perform either of the following tasks:
   - Type the path and name of the file.
   - Click **Browse** to navigate to the file, and select the file.
2. When you have finished uploading the files, click **Upload**.

## Setting the password policy

You can set up a password policy for the hub community, if you want to use values other than those set (by the system) as defaults. The password policy applies to all users who log in to the Community Console.

You can change the following elements of the password policy:

- Minimum Length, which represents the minimum number of characters the partner must use for the password. The default is 8 characters.
- Expire Time, which represents the number of days until the password expires. The default is 30 days.
- Uniqueness, which specifies the number of passwords to be held in a history file. A partner cannot use an old password if it exists in the history file. The default is 10 passwords.
- Special Characters, which, when selected, indicates that passwords must contain at least three of the following types of special characters:
  - Uppercase characters
  - Lowercase characters
  - Numeric characters
  - Special characters

  This setting allows for stricter security requirements when passwords are composed of English characters (ASCII). The default setting is off. It is recommended that Special Characters remain off when passwords are composed of international characters. Non-English-language character sets might not contain the required three out of four character types.

  The special characters supported by the system are as follows: '#', '@', '$', '&', '+'.
- Name Variation Checking, which, when selected, prevents the use of passwords that comprise an easily guessed variation of the user’s login or full name. This field is selected by default.

To change the default values:

1. Click **Hub Admin > Console Configuration > Password Policy**. The Password Policy page is displayed.
2. Click the **Edit** icon.
3. Change any of the default values to the ones you want to use for your password policy.

4. Click **Save**.

# Configuring permissions

Permissions represent privileges that a user must have to access various Console modules.

## How permissions are granted to users

Before you configure permissions, it is helpful to understand how permissions are granted to individual users. All three types of entities in the hub community–the Hub Administrator, the Internal Partner, and External Partners–can have an Admin user. When you create an Internal Partner or partner, you can also create the Admin user for that entity.

**Note:** In the case of the Hub Operator partner, two administrative users are automatically created at install time: an Admin user and the hubadmin user.

When you create the partner (as defined in "Creating partner profiles" on page 23), you provide the partner with login information (such as the name to use to log in and the password). After the partner logs in, the partner creates additional users within the organization. The partner also creates groups and assigns users to those groups. For example, an organization might want to have a group for people who monitor document volume. The partner would create a Volume group and add users to it.

**Note:** As a hub administrator, you can also define the users and groups for a partner.

The Admin user for the partner would then assign permissions to that group of users. For example, the Admin user might decide that the Volume group should see only the Document Volume and Document Analysis reports. The Admin user, using the Group Details page, would enable the document reports module but disable all other modules for the Volume group.

The setting you, as the hub administrator, make on the Permissions page determines whether a module is listed on the Group Details page.

Some modules are restricted to certain members of the hub community (for example, the hub administrators, like hubadmin). Therefore, even if you enable one of these modules for use by a partner, the module is not displayed on the Group Details page for the partner.

## Enabling or disabling permissions

### About this task

From the Permission List page, you can determine which permissions are available to assign to groups of users by enabling or disabling the permissions. You cannot, however, define new permissions.

To change the default permissions:

1. Click **Hub Admin > Console Configuration > Permissions**. The Permission List is displayed.
2. If you want to change the defaults, perform the following steps:
   a. Click the current setting (**Enabled** or **Disabled**) to change the setting.

b. When you are prompted to confirm the change, click **OK**.

## Setting console time out value

The default session timeout value of 30 minutes may not be acceptable during the following scenarios:

- Users in secure environments might need shorter session timeout periods to ensure security. This is also applicable whenever they leave their machine and forget to log off from the console.
- Users might need longer session timeout periods if they respond slower than typical users for accessibility reasons.

To set the timeout value for the WebSphere Partner Gateway console, perform the following steps:

1. Open WebSphere Application Server console
2. Navigate to **Servers** > **Application servers** > **bcgserver** > **Web Container Settings** > **Web container** > **Session management** .
3. In the **Session management** page, select **Set timeout** in **Session timeout** section.
4. Enter the value in minutes. The default is 30 minutes.
5. Click **Apply**

# Chapter 7. Defining receivers

This chapter describes how to set up receivers on WebSphere Partner Gateway. It covers the following topics:



**Note:** You should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Overview of receivers

As described in "Overview of document processing" on page 12, the *receiver* is responsible for accepting inbound documents from a specific transport. A receiver instance is configured for a particular deployment.

Documents received at a receiver on the hub can come from external partners (for eventual delivery to the internal partner) or from an internal partner back-end application (for eventual delivery to external partners).

Figure 16 on page 56 shows a WebSphere Partner Gateway server with four receivers set up. Two of the receivers (HTTP/S and FTP/S) are for documents coming from partners. These two receivers represent an HTTP URI and an FTP directory. You provide information about these receivers to your partners to indicate where they should send documents to you. The other two receivers (JMS and file directory) are for documents originating from the internal partner back-end application. These receivers represent a queue and a directory.

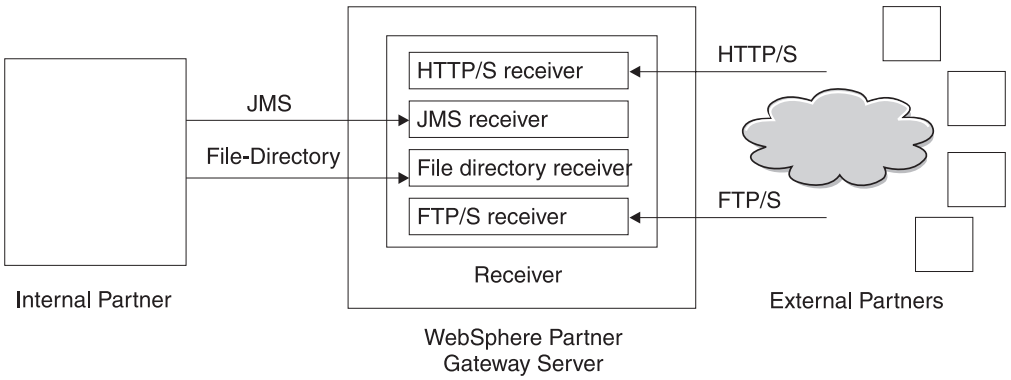



*Figure 16. Transports and associated receivers*

You set up at least one receiver for each type of transport over which documents will be sent to the hub. For example, you will have an HTTP receiver to receive any documents sent over the HTTP or HTTPS transport. If your external partners will be sending documents over FTP, you will set up an FTP receiver.

If you have special requirements for some documents that are received, you may need to set up more than one receiver for a given transport. In this case, you would tell your partners these requirements and ask them to send such documents to specific addresses so the correct receiver processing can be performed.

The Receiver component detects when a message arrives at one of the receivers. Some receivers detect messages by polling their transports at regular intervals or on a scheduled basis to determine if new messages have arrived. The WebSphere Partner Gateway receivers that are polling-based are: JMS, FTP, SMTP, File, and FTP Scripting. The HTTP/S receiver is callback-based, which means that it receives notification from the transport when messages arrive. User-defined transports can be either polling-based or callback-based.

## Uploading user-defined handlers

### About this task

You can modify configuration points for receivers by specifying a handler for the receiver. The handler can be supplied by WebSphere Partner Gateway or it can be a user-defined handler. This section describes how to upload a user-defined handler. Use this section only for user-defined handlers. Handlers supplied by WebSphere Partner Gateway are already available for use.

To upload a handler, perform the following steps:

1. From the main menu, click **Hub Admin > Hub Configuration > Handlers**.
2. Click **Receiver**.

   The list of handlers currently defined for receivers is displayed. Notice that handlers provided by WebSphere Partner Gateway have a Provider ID of **Product**.
3. From the Handler List page, click **Import**.
4. On the Import Handler page, specify the path to the XML file that describes the handler, or use **Browse** to search for that XML file.

After a handler is uploaded, you can use it to customize the configuration points of receivers.

## Generic preprocess handlers

The preprocess configuration handler is available on all types of receivers, but it is not applicable to SMTP receivers. The following table describes the attributes you can set for a Generic Preprocess handler:

*Table 2. Generic Preprocess Handler*

| Attributes | Description |
|---|---|
| From Packaging Name | This attribute indicates the packaging associated with the document. This value must match the packaging specified in the document definition. |
| From Packaging Version | This attribute indicates the version of the packaging specified in **From Packaging Name**. For example, if the document has a packaging of None, then this value will be N/A. |
| From Protocol Name | This attribute indicates the protocol associated with the document. This value must match the protocol specified in the document definition. |
| From Protocol Version | This attribute indicates the version of the protocol specified in **From Protocol Name**. |
| From Process Code | This attribute indicates the process (document type) associated with this document. This value must match the document type in the document definition. |
| From Process Version | This attribute indicates the version of the process specified in **From Process Code**. |
| METADICTIONARY | This attribute indicates the dictionary name to which the document definition is associated. This value must match the protocol specified in the From Protocol Name field. |
| METADOCUMENT | This attribute indicates the document definition name associated with this document. This value must match the process specified From Process Code field. |
| METASYNTAX | This attribute indicates the syntax of the document that will be processed in this receiver; allowed values are edilchg(EDI interchange) / xml / rod (flat-file). |
| ENCODING | This attribute indicates the character encoding of the document. The default value is ASCII. |
| BCG_BATCHDOCS | This attribute is set it to **ON** if you want the documents to be processed in a batch. |
| SenderId, ReceiverId | This attribute indicates the receiver ID, sender ID, which are the participants' business IDs as configured in their profiles. |

## Setting global transport values

### About this task

You set global transport attributes that apply to FTP Scripting receivers. If you are not defining FTP Scripting receivers, this section does not apply to you.

1. Click **Hub Admin > Hub Configuration > Receivers** to display the Receiver List.
2. Click **Global Transport Attributes** link.

3. If the default values are appropriate for your configuration, click **Cancel**. Otherwise, continue with the remaining steps in this section.
4. Click the **Edit** icon next to **Global Attributes Listed by Category**.
5. Review and, if necessary, change **FTP Scripting Transport** and **FTP Scripting - Receivers and Destinations** values.

   The FTP Scripting transport uses a locking mechanism that prevents more than one FTP Scripting instance from accessing the same receiver at the same time. When an FTP Scripting transport is ready to send documents, it requests this lock. Default values are supplied for such things as how long a receiver instance waits to obtain the lock and how many times it attempts to retrieve it if the lock is in use. You can use these default values or change them. To change one or more of the values, type the new value or values. You can change:

   - **FTP Scripting Transport** values
     - **Lock Retry Count**, which indicates how many times the receiver will attempt to obtain a lock if the lock is currently in use. The default is 3.
     - **Lock Retry Interval (Seconds)**, which indicates the amount of time that will elapse between attempts to obtain the lock. The default is 260 seconds.
   - **FTP Scripting - Receivers and Destinations** values
     - **Maximum Lock Time (Seconds)**, which indicates how long the receiver can hold the lock. The default is 240 seconds.
     - **Maximum Queue Age (Seconds)**, which indicates how long the receiver will wait in a queue to obtain the lock. The default is 740 seconds.
6. Click **Save**.

# Setting up an HTTP/S receiver

## About this task

The Receiver component has a predefined bcgreceiver servlet that is used to receive HTTP/S POST messages. You create one or more HTTP receivers to access the messages received by the servlet.

The following steps describe what you need to specify for an HTTP/S receiver.

1. Click **Hub Admin > Hub Configuration > Receivers** to display the Receiver List page.
2. From the Receiver List page, click **Create Receiver**.

# Receiver details

## About this task

In the **Receiver Details** section, perform the following steps:

## Procedure

1. Type a name for the receiver. For example, you might call the receiver HttpReceiver1. This is a required field. The name you enter here will be displayed on the Receivers list.
2. Optionally, indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.
3. Optionally, enter a description of the receiver.

4. Select **HTTP/S** from the **Transport** list.

## Receiver configuration

### About this task

In the **Receiver Configuration** section, perform the following steps:

1. Optionally, specify the operation mode. The operation mode defines the nature of the transmission. For example, if you want to test a document exchange before putting it into production, you would enter **Test**. The default is **Production**.
2. Enter the URI for the HTTP/S receiver. The name must begin with **bcgreceiver**. For example, you might enter /bcgreceiver/Receiver. Documents coming into the server over HTTP/S would then be received at /bcgreceiver/Receiver.
3. To authenticate a HTTP/S Receiver using the header attribute, set the **Enable basic authentication** flag to true. The default value is false.
4. Review and, if necessary, change the **HTTP/S Transport** values. You can change:
   - **Maximum Synchronous Timeout (Seconds)**, to indicate the number of seconds a synchronous connection can remain open. The default is 300 seconds.
   - **Maximum Simultaneous Synchronous Connections**, to indicate how many synchronous connections the system will allow. The default is 100 connections.

**Note:** You can edit the **Sync Routing** values.

## Handlers

If you will be receiving files containing multiple EDI interchanges or XML or ROD documents that need to be split, configure the appropriate splitter handler in the Preprocess configuration point.

If you will be sending or receiving certain types of business documents (RosettaNet, cXML, SOAP, and AS2) through a synchronous exchange, specify a handler for the associated protocol in the SyncCheck configuration point.

You can also modify the Postprocess configuration points for the receiver.

To modify a configuration point, go to " Modifying configuration points" on page 72. Otherwise, click **Save**.

# Setting up an FTP receiver

### About this task

An FTP receiver polls your FTP server at a set interval to look for new documents.

The following steps describe what you need to specify for an FTP receiver.

### Procedure

1. Click **Hub Admin > Hub Configuration > Receivers** to display the Receiver List page.
2. From the Receiver List page, click **Create Receiver**.

### Results

## Receiver details

### About this task

In the **Receiver Details** section, perform the following steps:

### Procedure

1. Type a name for the receiver. For example, you might call the receiver FTPReceiver1. This is a required field. The name you enter here will be displayed on the Receivers list.
2. Optionally indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.
3. Optionally enter a description of the receiver.
4. Select **FTP Directory** from the **Transport** list.

## Receiver configuration

### About this task

In the **Receiver Configuration** section, perform the following steps:

1. In the **FTP Root Directory** field, enter the root directory of the FTP server. The Document Manager automatically polls the partner subdirectories within the FTP root directory for document routing. This field is required. Refer to "Configuring the FTP server for receiving documents" on page 33for information about setting up the directory for an FTP server.

   **Note:** Type the directory path ending at the root FTP directory. Do not include the partner subdirectories.
2. Optionally, enter a value for **File Unchanged Interval** to indicate the number of seconds the file size must remain unchanged before the Document Manager retrieves the document for processing. This unchanged interval period ensures that a document has completed its transmission (and is not still in transit) when the Document Manager retrieves it. The default value is 3 seconds.
3. Optionally enter a value for **Number of Threads**, to indicate the number of documents the Document Manager can process simultaneously. The default value of 1 is recommended.
4. Optionally enter a value for **File Extensions to Exclude** to indicate the types of documents the Document Manager should ignore (exclude from processing) if it finds the documents in the FTP directory. For example, you might want the Document Manager to ignore spreadsheet files, in which case you would enter the extension associated with them. After you type the extension, click **Add**. The extension is then added to the list of file extensions to be ignored. The default is that no file types are excluded.

   **Note:** Do not use a dot preceding the file name extension (for example: .exe or . txt). Use only the characters that denote the file extension.

## Handlers

If you will be receiving files containing multiple EDI interchanges or XML or ROD documents that need to be split, configure the appropriate splitter handler in the Preprocess configuration point.

To modify the Preprocess configuration point, go to " Modifying configuration points" on page 72. Otherwise, click **Save**.

# Setting up an SMTP (POP3) receiver

## About this task

An SMTP receiver polls your POP3 mail server (according to the schedule you specify) to look for new documents.

The following steps describe what you need to specify for an SMTP (POP3) receiver.

## Procedure

1. Click **Hub Admin > Hub Configuration > Receiver** to display the Receiver List page.
2. From the Receiver List page, click **Create Receiver**.

## Results

# Receiver details

## About this task

In the **Receiver Details** section, perform the following steps:

## Procedure

1. Type a name for the receiver. For example, you might call the receiver POP3Receiver1. This is a required field. The name you enter here will be displayed on the Receivers list.
2. Optionally indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.
3. Optionally enter a description of the receiver.
4. Select **POP3** from the **Transport** list.

# Receiver configuration

## About this task

In the **Receiver Configuration** section of the page, perform the following steps:

## Procedure

1. Optionally indicate the Operation Mode. The Operation Mode defines the nature of the transmission. For example, if you want to test a document exchange before putting it into production, you would enter **Test**. The default is **Production**.
2. Enter the location of the POP3 server where mail is delivered. For example, you might enter an IP address.
3. Optionally enter a port number. If you do not enter anything, the value of 110 is used.
4. Enter the user ID and password required to access the mail server, if a user ID and password are required.

5. The **Number of Threads** is in read-only mode. This indicates that the number of documents the Document Manager can process simultaneously.

## Schedule

### About this task

In the **Schedule** section of the page, perform the following steps:

1. Select **Interval Based Scheduling** or **Calendar Based Scheduling**.
2. Perform one of the sets of steps:
   - If you select **Interval Based Scheduling**, select the number of seconds that should elapse before the POP3 server is polled again (or accept the default value). If you select the default value, the POP3 server is polled every 5 seconds.
   - If you select **Calendar Based Scheduling**, choose the type of scheduling (**Daily Schedule**, **Weekly Schedule**, or **Custom Schedule**).
     - If you select **Daily Schedule**, select the time of day (the hours and minutes) when the POP3 server should be polled.
     - If you select **Weekly Schedule**, select one or more days of the week in addition to the time of day.
     - If you select **Custom Schedule**, select the time of day and then choose **Range** or **Selective Days** for the week and the month. With **Range**, you indicate the start date and the end date. (For example, you can click **Mon** and **Fri** if you want the server polled at a certain time on weekdays only.) With **Selective Days**, you choose the specific days of the week and month.

## Handlers

If you will be receiving files containing multiple EDI interchanges or XML or ROD documents that need to be split, configure the appropriate splitter handler in the Preprocess configuration point.

To modify the Preprocess configuration point, go to " Modifying configuration points" on page 72. Otherwise, click **Save**.

# Setting up a JMS receiver

### About this task

A JMS receiver polls a JMS queue (according to the schedule you specify) to look for new documents.

The following steps describe what you need to specify for a JMS receiver.

1. Click **Hub Admin > Hub Configuration > Receivers** to display the Receiver List page.
2. From the Receiver List page, click **Create Receiver**.

**Note:** For information on configuring the runtime libraries so that the requisite WebSphere MQ jar files are visible to WebSphere Partner Gateway, see "Configuring the runtime libraries" on page 40.

## Receiver details

### About this task

In the **Receiver Details** section, perform the following steps:

1. Type a name for the receiver. For example, you might call the receiver JMSReceiver1. This is a required field. The name you enter here will be displayed on the Receiver list.
2. Optionally indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.
3. Optionally enter a description of the receiver.
4. Select **JMS** from the **Transport** list.

## Receiver configuration

### About this task

In the **Receiver Configuration** section of the page, perform the following steps:

1. Optionally indicate the **Operation Type**. The Operation Type defines the nature of the transmission. For example, if you want to test a document exchange before putting it into production, you would enter **Test**. The default is *Production*.
2. Enter the **JMS Provider URL**. This should match the value you entered (the file system path to the bindings file) when you configured WebSphere Partner Gateway for JMS (step 5 on page 38). You can also specify the subfolder for the JMS context as part of the JMS provider URL.

   For example, without the JMS context, you would enter `c:/temp/JMS`. With the JMS context, you would enter `c:/temp/JMS/JMS`.
3. Enter the **User ID** and **Password** required to access the JMS queue, if a user ID and password are required.
4. Enter a value for **JMS Queue Name**. This is a required field. This name should match the one you specified with the `define q` command when you created the bindings file (step 4 on page 39).

   If you entered the subfolder for the JMS context in step 2, enter only the queue name here (for example, `inQ`). If you did not enter the subfolder for the JMS context in the JMS provider URL, specify the subfolder before the factory name (for example, `JMS/inQ`).
5. Enter a value for the **JMS Factory Name**. This is a required field. This name should match the one you specified with the `define qcf` command when you created the bindings file (step 4 on page 39).

   If you entered the subfolder for the JMS context in step 2, enter only the factory name here (for example, `Hub`). If you did not enter the subfolder for the JMS context in the JMS provider URL, specify the subfolder before the factory name (for example, `JMS/Hub`).
6. Optionally enter the **Provider URL Package**.
7. Enter the **JNDI Factory Name**. This is a required field. The value of com.sun.jndi.fscontext.RefFSContextFactory is the one you will probably use, if you set up your JMS configuration for WebSphere MQ as described in "Configuring the hub for the JMS transport protocol" on page 37.
8. Enter the **JMS User Name** and **JMS Password**.
9. Optionally enter a value for **Time Out**, to indicate the number of seconds the receiver will monitor the JMS queue for documents. This field is optional.

10. Optionally enter a value for **Number of Threads**, to indicate the number of documents the Document Manager will process simultaneously. The default value of 1 is recommended.

For example, if you wanted to set up a JMS receiver to match the JMS configuration example in "Configuring the hub for the JMS transport protocol" on page 37, you would:

1. Enter the value **JMSReceiver** in the **Receiver Name** box.
2. Enter either of the following value in the**JMS Provider URL** box:
   - **file:///C:/TEMP/JMS/JMS** in case of windows.
   - **file:///opt/temp** in case of UNIX.
3. Enter the value **inQ** in the **JMS Queue Name** box.
4. Enter the value **Hub** in the **JMS Factory Name** box.

## Handlers

If you will be receiving files containing multiple EDI interchanges or XML or ROD documents that need to be split, configure the appropriate splitter handler in the Preprocess configuration point.

To modify configuration points for this receiver, go to " Modifying configuration points" on page 72. Otherwise, click **Save**.

# Setting up a File Directory receiver

### About this task

A File directory receiver polls a directory according to a set interval to look for new documents.

The following steps describe what you need to specify for a file directory receiver.

1. Click **Hub Admin > Hub Configuration > Receivers** to display the Receiver List page.
2. From the Receiver List page, click **Create Receiver**.

## Receiver details

### About this task

In the **Receiver Details** section, perform the following steps:

1. Type a name for the receiver. For example, you might call the receiver FileReceiver1. This is a required field. The name you enter here will be displayed on the Receiver list.
2. Optionally indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.
3. Optionally enter a description of the receiver.
4. Select **File Directory** from the **Transport** list.

## Receiver configuration

### About this task

In the **Receiver Configuration** section of the page, perform the following steps:

1. Enter a value for **Document Root Path** to indicate where the documents will be received.

   If the root directory does not exist, then a new directory is created for the receiver. But, if the root directory already exists, then the existing directory will be used by the receiver. This is applicable only from WebSphere Partner Gateway 6.1.1 onwards.

   The `file://` prefix is optional.

   For example, if you want to specify the directory `c:\wpg\receivers\file1` as the Document Root Path, enter `c:\wpg\receivers\file1` or `file://c:\wpg\receivers\file1`.
2. Optionally enter a value for **Poll Interval** to indicate how often the directory should be polled for new documents. If you do not enter anything, the directory will be polled every 5 seconds.
3. Optionally, enter a value for **File Unchanged Interval** to indicate the number of seconds the file size must remain unchanged before the Document Manager retrieves the document for processing. This unchanged interval period ensures that a document has completed its transmission (and is not still in transit) when the Document Manager retrieves it. The default value is 3 seconds.
4. Optionally enter a value for **Number of Threads**, to indicate the number of documents the Document Manager can process simultaneously. The default value of 1 is recommended.

### Handlers

If you will be receiving files containing multiple EDI interchanges or XML or ROD documents that need to be split, configure the appropriate splitter handler in the Preprocess configuration point.

To modify the Preprocess configuration point, go to " Modifying configuration points" on page 72. Otherwise, click **Save**.

## Setting up an FTP Scripting receiver

### About this task

An FTP Scripting receiver is a polling receiver that runs according to the schedule you set. The behavior of an FTP Scripting receiver is governed by an FTP command script.

Unlike the FTP receiver, which polls a directory on your FTP server, the FTP Scripting receiver polls directories on another server (for example, a VAN).

**Note:**

1. If the database is down and lock User is set to "Yes," the FTP Scripting Receiver might not work because it will not get the lock from database.
2. The partner needs to ensure the document is complete for the FTP Scripting Receiver to receive it. This can be done either by having the FTP Server keep the document locked until the document is complete or by having the partner write the document to a temporary directory and then move the completed document to the directory being used by the FTP Scripting Receiver.

## Creating the FTP script

### About this task

The FTP servers can have specific requirements for the commands they will accept. To use an FTP Scripting receiver, you create a file that includes all the FTP commands required by the FTP server to which you are connecting. (You must receive this information from the administrator of the FTP server.)

1. Create a script for the receivers, to indicate the actions you want performed. The following script is an example of connecting to the specified FTP server (with the name and password specified), changing to the specified directory on the FTP server, and receiving all the files in that directory:

   ```
   open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
   cd %BCGOPTION1%
   mget *
   quit
   ```

   The placeholders (for example, %BCGSERVERIP%) are replaced when the receiver is put in service by the values you enter when you create a specific instance of an FTP Scripting receiver. The %BCGOPTION% in this example is the name of the directory in the `cd` command. The script parameters and their associated FTP Scripting Receiver fields are shown in Table 3:

*Table 3. How script parameters map to FTP Scripting receiver field entries*

| Script parameter | FTP Scripting receiver field entry |
|---|---|
| %BCGSERVERIP% | Server IP |
| %BCGUSERID% | User ID |
| %BCGPASSWORD% | Password |
| %BCGOPTIONx% | Option*x*, under **User defined attributes** |

2. Save the file.

## FTP scripting commands

You can use the following commands when creating the script:

- ascii, binary, passive, epsv

  These commands are not sent to the FTP Server. They modify the mode of transfer (ascii, binary, or passive) to the FTP Server.
- cd

  This command changes to the specified directory.
- delete

  This command removes a file from the FTP server.
- get

  This command takes a single argument -- the name of the file to retrieve from the remote system. The requested file is then transferred into WebSphere Partner Gateway. Use this command only if you are picking up a single file and the name is known; otherwise, the `mget` command with wildcards should be used.
- getdel

  This command is the same as the `get` command, except that the file is removed from the remote system when WebSphere Partner Gateway gets the file for processing.
- mget

This command takes a single argument, which describes a group of files to be retrieved. The description can include the standard wildcard characters ('*' and '?'). One or more files are then retrieved from the remote system.

- mgetdel

  This command takes a single argument, which describes a group of files to be retrieved and then deleted from the FTP server. The description can include the standard wildcard characters (* and ?). One or more files are retrieved and then deleted from the remote system.

- mkdir

  This command creates a directory on the FTP server.

- mputren

  This command is a combination of mput and rename command. For example, **mputren * *.tmp /destination/*** command copies the file from the destination to FTP server with the extension **.tmp**. After the document download process is complete, the file is renamed and copied to **/destination** directory on the FTP root.

- open

  This command takes three parameters--the FTP server IP address, the user name, and a password. These parameters map to the %BCGSERVERIP%, %BCGUSERID%, and %BCGPASSWORD% variables.

  The first line of your FTP Scripting receiver script, therefore, should be:

  ```
  open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
  ```

- quit

  This command ends an existing connection to an FTP Server.

- quote

  This command indicates that everything after the QUOTE should be sent to the remote system as a command. This allows you to send commands to a remote FTP server that might not be defined in the standard FTP protocol.

- rename

  This command renames a file on the FTP server.

- rmdir

  This command removes a directory from the FTP server.

- site

  This command can be used to issue site-specific commands to the remote system. The remote system determines if the contents of this command are valid.

# Receiver details

## About this task

The following steps describe what you need to specify for an FTP Scripting receiver.

1. Click **Hub Admin > Hub Configuration > Receivers** to display the Receiver List page.
2. From the Receiver List page, click **Create Receiver**.

In the **Receiver Details** section, perform the following steps:

1. Type a name for the receiver. For example, you might call the receiver FTPScriptingReceiver1. This is a required field. The name you enter here will be displayed on the Receiver list.

2. Optionally indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.
3. Optionally enter a description of the receiver.
4. Select **FTP Scripting** from the Transport list.

## Receiver configuration

### About this task

In the **Receiver Configuration** section of the page, perform the following steps:

### Procedure

1. Optionally indicate the **Operation Mode**. The Operation Type defines the nature of the transmission. For example, if you want to test a document exchange before putting it into production, you would enter **Test**. The default is **Production**.
2. Enter the **Server IP** address of the FTP server to which you are connecting. The value you enter here will replace %BCGSERVERIP% when the FTP script is run.
3. Enter the **User ID** and **Password** you use to access the server. The values you enter here will replace %BCGUSERID% and %BCGPASSWORD% when the FTP script is run.
4. For **FTPS Mode**, select *Yes* or *No* to indicate whether the receiver will operate in secure sockets layer (SSL) mode. If Yes, then you will need to exchange certificates with your partners as described in Chapter 13, "Enabling security for document exchanges," on page 235.
5. Upload the script file by following these steps:
   a. Click **Upload Script File**.
   b. Type the name of the file that contains the script for processing documents, or use **Browse** to navigate to the file.
   c. Select the **Script File Encoding Type**.
   d. Click **Load File** to load the script file into the **Currently loaded script file** text box.
   e. If the script file is the one you want to use, click **Save**.
   f. Click **Close Window**.
6. For **Connection Timeout**, enter the number of seconds a socket will remain open with no traffic.
7. In the **Lock User** field, indicate whether the receiver will request a lock, so that no other instances of an FTP Scripting receiver can gain access to the same FTP server directory at the same time.

### Results

**Note:** The **Global FTP Scripting Attributes** values are already filled in, and you cannot edit them from this page. To modify these values, you use the Global Transport Attributes page, as described in " Setting global transport values" on page 57.

## User-defined attributes

### About this task

If you want to specify additional attributes, perform the following steps. The value you enter for the option will replace %BCGOPTION*x*% when the FTP script is run (where *x* corresponds to the number of the option).

1. Click **New**.
2. Type a value next to **Option 1**.
3. If you have additional attributes to specify, click **New** again and type a value.
4. Repeat step 3 as often as necessary to define all the attributes.

For example, suppose your FTP script looked like this:

```
open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
          cd %BCGOPTION1%
          mget *
          quit
```

The %BCGOPTION% in this case would be a directory name.

## Schedule

Indicate whether you want interval-based scheduling or calendar-based scheduling.

- If you select **Interval Based Scheduling**, select the number of seconds that should elapse before the FTP server is polled (or accept the default value).
- If you select **Calendar Based Scheduling**, choose the type of scheduling (**Daily Schedule**, **Weekly Schedule**, or **Custom Schedule**).
  - If you select **Daily Schedule**, enter the time of day at which the FTP server should be polled.
  - If you select **Weekly Schedule**, select one or more days of the week in addition to the time of day.
  - If you select **Custom Schedule**, select the time of day and then choose **Range** or **Selective Days** for the week and the month. With **Range**, you indicate the start date and the end date. (For example, you can click **Mon** and **Fri** if you want the server polled at a certain time on weekdays only.) With **Selective Days**, you choose the specific days of the week and month.

## Handlers

If you will be receiving files containing multiple EDI interchanges or XML or ROD documents that need to be split, configure the appropriate splitter handler in the Preprocess configuration point.

To modify the Preprocess configuration point, go to " Modifying configuration points" on page 72. Otherwise, click **Save**.

# Setting up a SFTP receiver

### About this task

This section provides details on using SFTP (SSH-FTP) as the protocol to transfer business documents. It provides confidentiality, authentication, and message integrity for the data.

The SFTP Receiver polls the SFTP Server, retrieves files from the SFTP server, and stores them in the local directory. The directory in the SFTP server that is polled is called the remote event directory. The directory in which the retrieved files are stored is called the local event directory. The following steps describe what you need to specify for a SFTP receiver.

The following steps describe what you need to specify for a SFTP receiver.

1. Click **Hub admin > Hub configuration > Receivers** to display the Receiver List page.
2. From the Receiver List page, click **Create Receiver**.

## Creating SFTP receiver on the WAS Administrative security enabled systems

### About this task

WebSphere Partner Gateway V6.2.1 facilitates creating SFTP Receiver on WAS Administrative security enabled systems. This topic details a task to create SFTP Receiver on a WAS Administrative security enabled system:

1. In WebSphere Partner Gateway Console, go to **System Administration → Console Administration → WAS Admin Security**.
2. On this screen, set the value of **bcg.RMIConnector.security.enabled** attribute to true. Note that the value of this attribute is "false" by default.
3. Set the remaining attributes on this screen as illustrated in the following procedure:
   a. **bcg.RMIConnector.security.enabled**: Set this attribute to "true" only if you have **WAS Admin Security** enabled. If you do not set this property to "true", you will not be able to create SFTP Receiver.
   b. **bcg.RMIConnector.security.enabled**: If this attribute is set to "true", then the following attributes need to be mandatorily set:
      - **bcg.RMIConnector.host.name**: Enter the hostname or the IP address of the Deployment manager.
      - **bcg.RMIConnector.portNumber**: Provide the BOOTSTRAP PORT of the Deployment manager.
      - **bcg.RMIConnector.admin.userId**: Set this attribute to user ID that is used for WAS Administrative security.
      - **bcg.RMIConnector.admin.password**: Set this attribute to password that is used for WAS Administrative security.
4. Click **Save**.

## Receiver details

### About this task

In the **Receiver Details** section, perform the following steps:

### Procedure

1. Type a name for the receiver. For example, you might call the receiver SFTPReceiver1. This is a required field. The name you enter here will be displayed on the Receivers list.
2. Optionally indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.

3. Optionally enter a description of the receiver.
4. Select **SFTP** from the **Transport** list.

## Receiver configuration

### About this task

In the **Receiver Details** section, perform the following steps:

### Procedure

1. Enter the **Operation Mode**. Select from the drop-down list or click on **New** to create a mode.
2. In the **SFTP Host IP** field, enter the hostname of SFTP Server. It will accept a maximum of 100 characters. You can also enter IP addresses, IPv4, and IPv6 addresses.
3. Enter the value of the **Port Number**. The default value is 22.
4. **Remote Event Directory** is the directory from which the adapter downloads event files from the SFTP site.
5. In the **Authentication Type**, select **User Name / Password** or **Private Key** authentication.
6. Enter the **User Id** and **Password** for username/password. If the Authentication type is private key authentication, then enter the User Name, Private key file, and Pass Phrase. The private key file has to be in OpenSSH format.
7. In the **SFTP Poll Interval**, enter the length of time in milliseconds. It is the time duration for which the adapter waits while polling the local event directory. This length of time and the length of time to process the documents in the local event directory is called the poll cycle.
8. **Poll Quantity** is the number of events (documents) that the receiver processes during each poll cycle.
9. **Retry Interval** is the length of time in milliseconds for which the adapter waits between attempts to establish a new connection after an error during inbound operations.
10. **Retry Limit** is the number of times the adapter tries to reestablish an inbound connection after an error.
11. **EIS encoding** is the encoding of the FTP server. Use this value to set the encoding for the control connection to the FTP server.
12. The **Enable server authentication** can be enabled to authenticate the server to which the connection is being established. If the server authentication is enabled, enter the host key file path. The host key file has to be in OpenSSH format.
13. Configure handlers, if required
14. Click **Save** to save the configuration.

## Handlers

If you will be receiving files containing multiple EDI interchanges or XML or ROD documents that need to be split, configure the appropriate splitter handler in the Preprocess configuration point.

To modify the Preprocess configuration point, go to " Modifying configuration points" on page 72. Otherwise, click **Save**.

## Setting up a receiver for a user-defined transport

### About this task

If you are defining a receiver for a user-defined transport, the field names and other information are defined within the file that describes the transport.

Perform the following steps:

1. Click **Hub Admin > Hub Configuration > Receiver**.
2. Click **Manage Transport Types**.
3. Enter the name of an XML file that defines the transport (or use **Browse** to navigate to the file).
4. Click **Upload**.

   **Note:** From the Receiver List, you can also delete a user-defined transport type. You cannot delete a transport provided by WebSphere Partner Gateway. Also, you cannot delete a user-defined transport after it has been used for creating a receiver.
5. Click **Create Receiver**.
6. Type a name for the receiver. This is a required field. The name you enter here will be displayed on the Receiver list.
7. Optionally indicate the status of the receiver. **Enabled** is the default. A receiver that is enabled is ready to accept documents. A receiver that is disabled cannot accept documents.
8. Optionally enter a description of the receiver.
9. Select the user-defined transport from the list.
10. Complete the fields (which will be unique for each user-defined transport).
11. If you want to modify configuration points for this receiver, go to " Modifying configuration points." Otherwise, click **Save**.

## Modifying configuration points

The number of configuration points available and the number of associated handlers for those configuration points vary, depending on the type of receiver you are setting up. For example, the SyncCheck configuration point is available only with HTTP/S and JMS receivers.

For certain business protocols (RosettaNet, cXML, SOAP, and AS2) involved in synchronous exchanges, you must specify a handler for that protocol in the SyncCheck configuration point. You can also modify the way receivers process documents by applying an uploaded user-defined handler (or a product-provided process) to the Preprocess and Postprocess points of the receiver.

To apply a user-written handler for these configuration points, you must first upload the handler, as described in " Uploading user-defined handlers" on page 56. You can also use a product-provided handler, which is already available and does not have to be uploaded.

### Preprocess

The Preprocess configuration handler is available on all types of receivers but is not applicable to SMTP receivers.

## Preprocess attributes

Table 4 describes the attributes you can set for a Preprocess handler and lists the splitter handlers to which the attributes apply.

The ROD attributes used as examples in this table correspond to those used in " ROD to EDI example" on page 342. In the example, the ROD attributes are contained in the map S_DT_ROD_TO_EDI.eif, which includes the following document definition:

- Package: None (version N/A)
- Protocol: ROD_TO_EDI_DICT (version ALL)
- Document Type: DTROD-TO-EDI_ROD (version ALL)

The ROD metadictionary and metadocument associated with this flow are ROD_TO_EDI_DICT and DTROD-TO-EDI_ROD.

*Table 4. Splitter handler attributes*

| Attribute | Description | Splitter Handler |
|---|---|---|
| Encoding | The character encoding of the document. The default is ASCII. | ROD Generic XML EDI |
| BATCHDOCS | When BCG_BATCHDOCS is on, the splitter adds batch IDs to the documents after the documents are split. If the documents are transformed into EDI transactions to be enveloped, the Enveloper uses the batch IDs to make sure that the transactions are put into the same EDI interchange (if possible) before being delivered. Note that the Enveloper must have the batching attribute set to **On** (the default value). See "Batch mode" on page 180. | ROD Generic XML |
| From Packaging Name | The packaging associated with the document. This value must match the packaging specified in the document definition. For example, for a document that has a packaging of None, this value should be **None**. | ROD Generic |
| From Packaging Version | The version of the packaging specified in From Packaging Name. For example, if the document has a packaging of None, this value would be **N/A**. | ROD Generic |
| From Protocol Name | The protocol associated with the document. This value must match the protocol specified in the document definition. For example, for a ROD document, this value could be **ROD-TO-EDI_DICT**. | ROD Generic |
| From Protocol Version | The version of the protocol specified in From Protocol Name. For example, for the ROD-TO-EDI_DICT protocol, the value would be **ALL**. | ROD Generic |
| From Process Code | The process (document type) associated with this document. This value must match the document type in the document definition. For example, for a ROD document, this value could be DTROD-TO-EDI_ROD. | ROD Generic |
| From Process Version | The version of the process specified in From Process Code. For example, for DTROD-TO-EDI_ROD, this value would be **ALL**. | ROD Generic |
| Metadictionary | The metadictionary provides information that lets WebSphere Partner Gateway interpret the data. For example, for a ROD document, this value could be **ROD-TO-EDI_DICT**. | ROD Generic |

*Table 4. Splitter handler attributes (continued)*

| Attribute | Description | Splitter Handler |
|---|---|---|
| Metadocument | The metadocument provides information that lets WebSphere Partner Gateway interpret the data. For example, for a ROD document, this value could be **DTROD-TO-EDI_ROD**. | ROD Generic |
| Metasyntax | The metasyntax describes the format of the document being split. The default value is **rod**. | ROD Generic |
| SenderId | The Sending Partner's ID. | Generic |
| ReceiverId | The Receiving Partner's ID. | Generic |

**Notes:**

1. Only one ROD document type per receiver instance is supported.
2. If a receiver has more than one splitter handler configured (for example, if it has ROD, XML, and EDI splitter handlers configured), the ROD splitter handler must be the last one in the **Configured List**.

## Modifying the preprocess configuration point

### About this task

To modify the Preprocess configuration point, perform the following steps:

1. Select **Preprocess** from the **Configuration Point Handlers** list.

   Five preprocessing handlers are provided (by default) and are shown in the **Available List**.

   - com.ibm.bcg.edi.receiver.preprocesshandler.EDISplitterHandler
   - com.ibm.bcg.edi.receiver.preprocesshandler.XMLSplitterHandler
   - com.ibm.bcg.edi.receiver.preprocesshandler.RODSplitterHandler
   - com.ibm.bcg.edi.receiver.preprocesshandler.GenericDocumentFlowHandler
   - com.ibm.bcg.server.receiver.preprocesshandler.FileNamePartnerId

   **Note:** The preprocessing handlers do not apply to SMTP receivers.
2. If you will be receiving multiple EDI interchanges or XML or ROD documents that need to be split, make sure you select the appropriate splitter handler. To configure the Preprocess step:
   a. Select a handler from the **Available List** and click **Add**. Note that the handler moves from the **Available List** to the **Configured List**, as illustrated in Figure 17 on page 75:

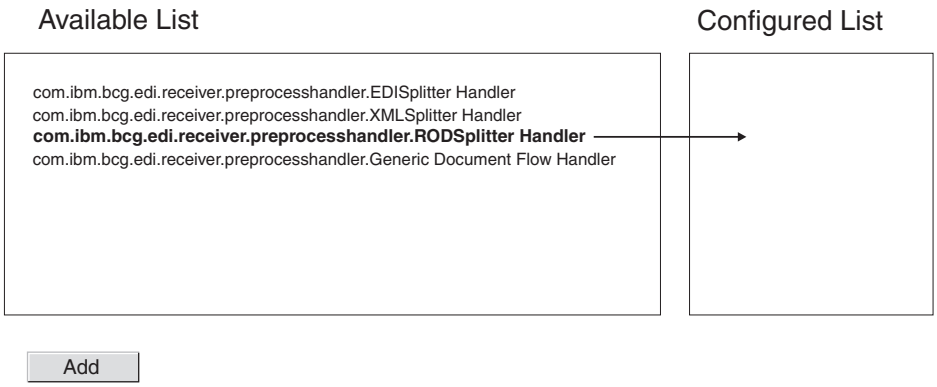



*Figure 17. Configuring the preprocessing step for a receiver*

b. Repeat this step for each handler you want to add to the configured list.

   Remember that for receivers, the handlers are called in the order in which they appear on the **Configured List**. The first applicable handler processes the request, and subsequent handlers on the list are not called.

c. Configure the handler by selecting it and clicking **Configure**:

   - If you have added the EDISplitterHandler, you can modify its attribute-Encoding. The default for encoding is ASCII.
   - If you have added the XMLSplitterHandler, you can modify its attribute--BCGBATCHDOCs. The default is **ON**. See "Preprocess attributes" on page 73 for information about this attribute.
   - If you have added the RODSplitterHandler, you can specify values for 11 attributes. Encoding, BATCHDOCS, and Metasyntax have default values. For the other attributes, you must type a value for From Packaging Name, From Packaging Version, From Protocol Name, From Protocol Version, From Process Code, From Process Version, Metadictionary, and Metadocument. See "Preprocess attributes" on page 73 for information about these attributes.
   - If you have added the GenericDocumentFlowHandler, you can specify values for 13 attributes. Encoding and BATCHDOCS have default values. The SenderId and ReceiverId attributes are pre-configured for the GenericDocumentFlowHandler without any default values. For the other attributes, you must type a value for From Packaging Name, From Packaging Version, From Protocol Name, From Protocol Version, From Process Code, From Process Version, Metadictionary, Metadocument, and Metasyntax. See "Preprocess attributes" on page 73 for information about these attributes.
   - If you have added the FileNamePartnerId, then it does not expect any configuration parameters. It expects the received file to follow this naming convention:

     ```
     <anystring>bcgrcv<Receiver ID>bcgsdr<Sender ID>bcgend<anystring>
     ```

     where

     *Receiver ID ,Sender ID*
     Are the participants' business IDs as configured in their profiles.

     **bcgrcv, bcgsdr**
     Are string constants that signal the start of the receiver and sender business IDs.

**bcgend**
Is a string constant that determines the end of the required naming convention string

*anystring*
is any alphanumeric character chosen by the user

This handler is can be configured only for FTP scripting or File directory receivers. To receive binary files over FTP Scripting or File directory, you can configure this handler for the receiver.

# SyncCheck

## About this task

The SyncCheck configuration point is available for HTTP/S and JMS receivers only.

To specify a handler for a business protocol involved in a synchronous exchange, perform the following steps:

1. Select **SyncCheck** from the **Configuration Point Handlers** list.

   Six SyncCheck handlers are provided (by default) for an HTTP/S receiver. These handlers are shown in the **Available List**:

   - com.ibm.bcg.server.sync.As2SyncHdlr
   - com.ibm.bcg.server.sync.CxmlSyncHdlr
   - com.ibm.bcg.server.sync.RnifSyncHdlr
   - com.ibm.bcg.server.sync.SoapSyncHdlr
   - com.ibm.bcg.server.sync.DefaultAsynchronousSyncCheckHandler
   - com.ibm.bcg.server.sync.DefaultSynchronousSyncCheckHandler
   - com.ibm.bcg.server.sync.EBMSSyncCheckHandler

   For example, if you are configuring an HTTP/S receiver, the Available List looks like this:

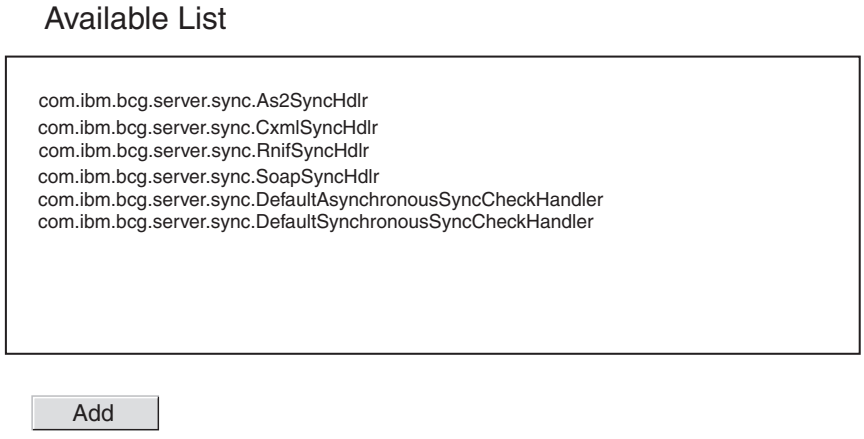



*Figure 18. List of available handlers for an HTTP/S SyncCheck configuration point*

As you can see from the naming convention, the first four handlers are specific to the four document types that can be used for synchronous transactions. Any request that uses the DefaultAsynchronousSyncCheckHandler will be treated as

an asynchronous request. Any request that uses the DefaultSynchronousSyncCheckHandler will be treated as a synchronous request.

The DefaultAsynchronousSyncCheckHandler and DefaultSynchronousSyncCheckHandler can be used with other receivers (such as a JMS receiver).

2. If you will be receiving synchronous documents at this receiver, perform the following steps:
   a. Select one or more of the handlers from the **Available List** and click **Add**.
   b. Repeat this step if you want to add other handlers to the list. Remember that for receivers, the handlers are called in the order in which they appear on the **Configured List**. The first available handler processes the request, and subsequent handlers on the list are not called.

   For HTTP and HTTPS receivers, it is a good practice to list the specific SyncCheck handler (for example, com.ibm.bcg.server.sync.As2SyncHdlr for AS2 transactions) before listing the default SyncCheck handlers.

## Postprocess

### About this task

No handlers are provided by default for the Postprocess step, and, therefore, no handlers are listed by default in the **Available List**. You can, however, upload a handler for this configuration point for all types of receivers that support synchronous communication. The available handler types for the postprocessing step are:

- RECEIVER.SYNCRESPONSEPROCESS.JMS
- RECEIVER.SYNCRESPONSEPROCESS.HttpS

You add a Postprocessing handler by uploading a handler that conforms to one of these handler types. You use the **Import** choice of the Handlers List page to upload a user-defined handler. When you upload a user-defined receiver handler, the handler is added to the Handlers List. It also appears on the Available List for the type of configuration point to which it pertains.

To modify the Postprocess configuration point, perform the following steps:

1. Select **Postprocess** from the **Configuration Point Handlers** list.
2. Select a user-defined handler from the **Available List** and click **Add**. Note that the handler moves from the **Available List** to the **Configured List**

## Modifying the configured list

### About this task

If you need to change the order of the handlers, delete a handler, or configure attributes for the handler, perform the appropriate step:

- Remove a handler by selecting the handler from the **Configured List** and clicking **Remove**. The handler is moved to the **Available List**.
- Rearrange the order in which the handler is used by selecting the handler and clicking **Move Up** or **Move Down**.
- Configure the handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured is displayed.

# Chapter 8. Configuring fixed workflow steps and actions

This chapter describes optional tasks you can perform to configure fixed inbound and outbound workflows and actions. If you do not need to change the product-provided behavior of workflows or actions, skip this chapter.

This chapter includes the following topics:



**Note:** You should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Uploading handlers

### About this task

If you are going to modify components, you first upload the handlers for those components before creating or configuring the components. You only need to upload the user-defined handlers for the components that require them. For example, if you are adding your own validation step, you need to upload that handler from the Actions page of **Handlers** (as described in steps 1 through 4).

**Note:** As mentioned in "Configuring document processing components with handlers" on page 14, you upload only user-defined handlers. The handlers supplied by WebSphere Partner Gateway are already available.

You can modify fixed workflows and actions and create new actions. You modify these components by the handlers you associate with them.

**Note:** You can list the valid handler types for actions and fixed workflows by clicking **Hub Admin > Hub Configuration > Handlers > Action > Handler Types** or **Hub Admin > Hub Configuration > Handlers > Fixed Workflow > Handler Types**. Use this list to confirm that your handler is a valid type before uploading it. It must be one of the allowed types or it will not upload successfully.

To upload a handler, perform the following steps:

1. From the main menu, click **Hub Admin > Hub Configuration > Handlers**.
2. Select the type of handler (**Action** or **Fixed Workflow**).

   The list of handlers currently defined for that particular component is displayed. Notice that handlers provided by WebSphere Partner Gateway are listed. They have a Provider ID of **Product**.
3. From the Handler List page, click **Import**.
4. On the Import Handler page, specify the path to the XML file that describes the handler, or use **Browse** to search for that XML file.
5. Click **Upload**.

After a handler is uploaded, you can use it to create new actions and workflows.

**Note:** You can update user-defined handlers by uploading the modified XML file. For an action handler, for example, you would click **Hub Admin > Hub Configuration > Handlers > Action**, and then click **Import**.

You cannot modify or delete handlers provided by WebSphere Partner Gateway.

# Configuring fixed workflows

## About this task

Chapter 2, "Introduction to hub configuration," on page 5 described the two fixed inbound workflow steps that you can configure--one for unpackaging a protocol and one for parsing the protocol. For outbound workflows, there is one step, for protocol packaging.

If you are going to use a user-defined handler to configure a workflow step, upload the handler, as described in " Uploading handlers" on page 79.

To configure a fixed workflow, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Fixed Workflow**.
2. Click either **Inbound** or **Outbound**.
3. Click the **View Details** icon next to the name of the step you want to configure.

   The step, along with a list of handlers already configured for that step, is listed. See "Inbound workflows" and "Outbound workflow" on page 81 for a list of default handlers.
4. Click the **Edit** icon to edit the list of handlers.
5. Perform one or more of the following tasks for each step you want to modify.
   a. Add a handler by selecting the handler from the **Available List** and clicking **Add**. (A handler appears in the **Available List** if you uploaded a user-defined handler or if you previously removed a handler from the **Configured List**.) The handler is moved to the **Configured List**.
   b. Remove a handler by selecting the handler from the **Configured List** and clicking **Remove**. The handler is moved to the **Available List**.
   c. Rearrange the order in which the handlers are called by selecting the handler and clicking **Move Up** or **Move Down**.

      Handlers are called in the order in which they are listed in the **Configured List**. The first available handler that can process the request is the one that handles the request. If you anticipate receiving a large number of documents of a certain type (for example, ROD documents), you can move the handler associated with that type of document (in this example, com.ibm.bcg.edi.business.process.RODScannerHandler) to the top of the list.
6. Click **Save**.

# Inbound workflows

This section lists the handlers configured for the inbound workflows.

## Protocol unpackaging handlers

By default, the Protocol Unpackaging step has the following handlers configured:

- com.ibm.bcg.ediint.ASUnpackagingHandler
- com.ibm.bcg.server.pkg.NullUnpackagingHandler

- com.ibm.bcg.server.pkg.MIMEMultipartUnpackagingHandler
- com.ibm.bcg.eai.EAIUnpackagingHandler

### Protocol processing handlers

By default, the Protocol Processing step has the following handlers configured:

- com.ibm.bcg.server.RNOChannelParseHandler
- com.ibm.bcg.server.RNSignalChannelParseHandler
- com.ibm.bcg.server.RNSCChannelParseHandler
- com.ibm.bcg.server.BinaryChannelParseHandler
- com.ibm.bcg.cxml.cXMLChannelParseHandler
- com.ibm.bcg.soap.SOAPChannelParseHandler
- com.ibm.bcg.server.XMLRouterBizProcessHandler
- com.ibm.bcg.edi.EDIRouterBizProcessHandler
- com.ibm.bcg.edi.business.process.RODScannerHandler
- com.ibm.bcg.edi.business.process.NetworkAckHandler
- com.ibm.bcg.server.EBMSProtocolParseHandler
- com.ibm.bcg.server.BackendChannelParseHandler

The "Content-Types" attribute is associated with BinaryChannelParseHandler, XMLRouterBizHandler, EDIRouterBizProcessHandler, and cXMLChannelParseHandler. These handlers are pre-populated with default list of content types. If the received document has a content type header that is configured for any of the above handlers, that handler gets applied.

## Outbound workflow

By default, the Protocol Packaging step has the following handlers configured:

- com.ibm.bcg.server.pkg.NullPackagingHandler
- com.ibm.bcg.ediint.ASPackagingHandler
- com.ibm.bcg.edi.server.EDITransactionHandler
- com.ibm.bcg.rosettanet.pkg.RNOPPackagingHandler
- com.ibm.bcg.server.pkg.RNPassThruPackagingHandler
- com.ibm.bcg.cxml.cXMLPackagingHandler
- com.ibm.bcg.soap.SOAPPackagingHandler
- com.ibm.bcg.eai.EAIPackagingHandler

# Configuring actions

Chapter 2, “Introduction to hub configuration,” on page 5 showed that actions can be made up of one or more steps. WebSphere Partner Gateway supplies a series of default actions. You can add to the list of actions by uploading one or more action handlers (which are steps in the action), which you can then use in an action. You can also create new actions, as described in “Creating actions” on page 97.

**Note:** You cannot modify the actions supplied by WebSphere Partner Gateway, although you can copy one of those actions and modify it, as described in “Copying an action” on page 98.

If you are going to use a user-defined handler to configure an action, upload the handler, as described in “ Uploading handlers” on page 79.

## Product-provided actions

This section provides details on the Websphere Partner Gateway product-supplied actions concerning their purpose and any configuration required to use them. Chapter 9, "Configuring document types," on page 99 provides more details on when to use some of these actions.

Some actions have Bi-Directional in their name. Here *Bi-Directional* means that either the source and target formats can be reversed and the action can still be used. For example, for the action "Bi-Directional Translation of RosettaNet and XML with Validation," the source document can be RosettaNet and the target document XML, or the source document can be XML and the target document RosettaNet.

Following are the various actions that are provided in Websphere Partner Gateway:

- "Pass Through"
- "Internal Partner Cancellation of RosettaNet Process" on page 83
- "RosettaNet Pass Through with Process Logging" on page 84
- "Bi-Directional Translation of RosettaNet and RosettaNet Service Content with Validation" on page 84
- "Bi-Directional Translation of RosettaNet and RosettaNet Service Content without Content Validation" on page 86
- "Bi-Directional Translation of Internal Partner Custom XML to RosettaNet with Content Duplicate Check and Validation" on page 86
- "Bi-Directional Translation of RosettaNet and XML with Validation" on page 85
- "Bi-Directional Translation of Custom XML with Validation" on page 87
- "Bi-Directional Translation of Custom XML with Duplicate Check and Validation" on page 88
- "Custom XML Pass Through with Duplicate Check and Validation" on page 89
- "Custom XML Pass Through with Duplicate Check" on page 89
- "Custom XML Pass Through with Validation" on page 90
- "EDI De-envelope" on page 90
- "EDI Validate and EDI Translate" on page 91
- "ROD (FlatFile) Translate and EDI Validate" on page 92
- "XML Translate and EDI Validate" on page 91
- "ebMS Split and Parse" on page 93
- "SOAP Envelope Validate" on page 95
- "SOAP Body validate" on page 95
- "SOAP De-envelope" on page 96
- "EDI interchange validate" on page 93
- "WTX Transformation" on page 93
- "EDI ReEnveloper" on page 95

### Pass Through

This action us used when no special processing, such as validation or transformation, is to be performed on the document. The source document is sent to the target location as it is.

### Configuration

None required.

### Modification

This action can be copied into a new action. New steps can be added before the existing steps. For example a custom validation step which validates the source document, or some other custom processing.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.passthrough.No_op** – Used to indicate the target document content type should not be derived from the document contents.
2. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Internal Partner Cancellation of RosettaNet Process

### Purpose

This action is meant for cancellation of a RosettaNet RNIF process by the internal partner (backend). When the backend application (internal partner) sends an XML Event document with the event code 800/801, then in this step a 0A1 document is created for sending to the external partner and the corresponding PIP process is cancelled.

### Configuration

The RNIF process that is being cancelled must have already been configured in WebSphere Partner Gateway and WebSphere Partner Gateway must have already received the RosettaNet document that started the process that is being cancelled.

### Modification

This action cannot be modified or copied because this action is specific to cancellation of the RosettaNet PIP process.

### Steps

This action contains the following steps, which are run in the sequence:

1. **com.ibm.bcg.server.pkg.UnPackagingFactory** - Determines correct unpackaging class for RNIF or assumes that the document is non-RNIF and no unpackaging takes place.
2. **com.ibm.bcg.validation.ValidationFactory** - Validates the source RN document for correct RNIF Service Content.
3. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## RosettaNet Pass Through with Process Logging

This action is used when the RosettaNet source RNIF document is pass through in WebSphere Partner Gateway. Use this step when the RNIF document Service Content is not getting extracted or transformed. Even though this is pass through there the RNIF processing is still performed with the Receipt Acknowledgments getting generated.

### Configuration

None required

### Modification

This action can be copied and modified. New steps can be added before the existing steps for additional custom processing.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.rosettanet.passthru.ProcessLoggingFactory** - This step sets RosettaNet document meta-data into the Business Document Object (BDO).
2. **com.ibm.bcg.passthrough.No_op** - Used to indicate the target document content type should not be derived from the document contents.
3. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Bi-Directional Translation of RosettaNet and RosettaNet Service Content with Validation

This action is used for RosettaNet RNIF documents. When receiving a RNIF document from the external partner the payload (RNSC - RosettaNet Service Content) will be extracted from the RNIF packaged document for sending to the backend application (internal partner). Validation will occur on the RNIF document including the RNSC. When coming from the backend application (internal partner) the RNSC document will be validated.

### Configuration

The RosettaNet PIP package for RosettaNet document has to have been loaded.

### Modification

This action cannot be copied and modified.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.server.pkg.UnPackagingFactory** - Determines correct unpackaging class for RNIF or assumes that the document is non-RNIF and no unpackaging takes place.
2. **com.ibm.bcg.validation.ValidationFactory** - Performs validation and makes use of the following BusinessProcesses for validating the RNIF 1.1, RNIF 2.0, and

RNSC documents. - RNSignal0A1Validation (validating the WebSphere Partner Gateway generated RNIF signals or 0A1 message) - ValidationNoOp (this just returns the BusinessDocument with out doing any processing, This will be called when WBIC does retries for RNIF signals or 0A1 message ) - RN11Validation (this is to validate the RNIF 1.1 message) - RN20Validation (this is to validate the RNIF 2.0 message) - RNSCValidation (this is to validate the XML event and RNSC message).

3. **com.ibm.bcg.translation.protocol.StdRNandRNSCProtFactory** - Used to extract the RNSC from the RNIF document or to create the RNIF information for the RNSC.
4. **com.ibm.bcg.sponsor.SponsorBusProcessFactory** - Used in processing RosettaNet 0A1 documents for updating the RosettaNet State Engine.
5. **com.ibm.bcg.outbound.OutboundDocFactor**y - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Bi-Directional Translation of RosettaNet and XML with Validation

This action is used for RosettaNet RNIF documents that need to be transformed to a Custom XML document, or vice versa. When receiving an RNIF document from the external partner the payload (RNSC - RNIF Service Content) will be extracted from RNIF packaging, validated and transformed into an XML document with the transformed target documents validated for sending to the backend application (internal partner). When coming from the backend application (internal partner) the XML will be validated, transformed into the RNSC that is validated.

### Configuration

- The RosettaNet PIP package for RosettaNet document has to have been loaded
- Requires the validation map (XML SCHEMA) to be configured on either the source or target XML document
- Requires an XSLT Transformation map to be configured for this action

### Modification

This action cannot be copied and modified.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.server.pkg.UnPackagingFactory** - Determines correct unpackaging class for RNIF or assumes that the document is non-RNIF and no unpackaging takes place.
2. **com.ibm.bcg.validation.ValidationFactory** – For validating the source RNIF or XML document.
3. **com.ibm.bcg.translation.protocol.RNXsltProtFactory** – Transforms the RNSC to / from the XML.
4. **com.ibm.bcg.validation.OutboundValidationFactory** - Validates the resultant transformed XML document.
5. **com.ibm.bcg.sponsor.SponsorBusProcessFactory**- Used in processing RosettaNet 0A1 documents for updating the RosettaNet State Engine.
6. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This

is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Bi-Directional Translation of RosettaNet and RosettaNet Service Content without Content Validation

This action is used for RosettaNet (RNIF) documents. On receiving a RNIF document from an external partner, the payload (RNSC - RNIF Service Content) is extracted from the RNIF packaging. The extracted payload is validated and transformed into an XML document before it is sent to the backend application (internal partner). When an XML document is received from the back end application (internal partner), the following steps are performed on it:

1. Duplicate ID check
2. Validation
3. Transformation into RNSC
4. Validation of RNSC

### Configuration

The RosettaNet PIP package for RosettaNet document has to have been loaded.

### Modification

This action cannot be copied and modified.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.server.pkg.UnPackagingFactory** - Determines correct unpackaging class for RNIF or assumes that the document is non-RNIF and no unpackaging takes place.
2. **com.ibm.bcg.validation.ValidationWithoutContentFactory** – Will perform validation except not on the RNSC.
3. **com.ibm.bcg.translation.protocol.StdRNandRNSCProtFactory** - Used to extract the RNSC from the RNIF document or to create the RNIF information for the RNSC.
4. **com.ibm.bcg.sponsor.SponsorBusProcessFactory** - Used in processing RosettaNet 0A1 documents for updating the RosettaNet State Engine.
5. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Bi-Directional Translation of Internal Partner Custom XML to RosettaNet with Content Duplicate Check and Validation

This action is used for RosettaNet RNIF documents that need to be transformed to a Custom XML document, or vice versa. When receiving a RNIF document from the external partner the payload (RNSC - RNIF Service Content) will be extracted from RNIF packaging, validated and transformed into an XML document for sending to the backend application (internal partner). When coming from the backend application (internal partner) a duplicate ID check will be done on the XML and then the XML will be validated, transformed into the RNSC and

validation performed. Similar to the " Bi-Directional Translation of RosettaNet and XML with Validation" action but with an additional Duplicate Check performed on the source XML.

### Configuration

- The source document XML Format needs the Duplicate Check Keys configured
- The RosettaNet PIP package for RosettaNet document has to have been loaded
- Requires the validation map (XML SCHEMA) to be configured on either the source or target XML document
- Requires an XSLT Transformation map to be configured for this action

### Modification

This action cannot be copied and modified, as this is specific to an RNIF document.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.duplicate.ContentDuplicateProcessFactory** - For a received Custom XML performs a duplicate ID check.
2. **com.ibm.bcg.server.pkg.UnPackagingFactory** - Determines correct unpackaging class for RNIF or assumes that the document is non-RNIF and no unpackaging takes place.
3. **com.ibm.bcg.validation.ValidationFactory** – For validating the source RNIF or XML document.
4. **com.ibm.bcg.translation.protocol.RNXsltProtFactory** – Transforms the RNSC to / from the XML.
5. **com.ibm.bcg.validation.OutboundValidationFactory** - Validates the resultant transformed XML document.
6. **com.ibm.bcg.sponsor.SponsorBusProcessFactory** - Used in processing RosettaNet 0A1 documents for updating the RosettaNet State Engine.
7. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Bi-Directional Translation of Custom XML with Validation

This action is used with custom XML documents coming from an external partner or the internal partners. The source document is validated, transformed into the target document and the target document is validated.

### Configuration

- Requires the validation map (XML SCHEMA) to be configured on the source document
- Requires an XSLT Transformation map to be configured for this action
- Requires the validation map (XML SCHEMA) to be configured on the target document

### Modification

This action can be copied and modified. The Transformation or Validation steps can be substituted with User defined steps or additional User defined steps added.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.validation.ValidationFactory** – This step validates the received Custom XML document.
2. **com.ibm.bcg.translation.protocol.translators.xslt.XSLTranslationFactory** – Performs the transformation.
3. **com.ibm.bcg.validation.OutboundValidationFactory** - Validates the resultant transformed XML document.
4. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Bi-Directional Translation of Custom XML with Duplicate Check and Validation

This action is used with custom XML documents. Can be used for documents coming from the external partner or the internal partner. Duplicate ID checking is performed on the source document, validation on the source document, transformation of the source document to the target document, and validation of the target document. This action is similar to the "Bi-Directional Translation of Custom XML with Validation" except with the additional Duplicate Check step.

### Configuration

- The source document XML Format needs the Duplicate Check Keys configured
- Requires the validation map (XML SCHEMA) to be configured on the source document
- Requires an XSLT Transformation map to be configured for this action
- Requires the validation map (XML SCHEMA) to be configured on the target document

### Modification

This action can be copied and modified. The steps that can be substituted with User defined steps are ValidationFactory, XSLTTranslationFactory and OutboundValidationFactory or additional User defined steps added.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.duplicate.ContentDuplicateProcessFactory** - Checks for a duplicate document based on the document ID.
2. **com.ibm.bcg.validation.ValidationFactory** - This step validates the received Custom XML document.
3. **com.ibm.bcg.translation.protocol.translators.xslt.XSLTranslationFactory** - This step transforms the receiving Custom XML document to target XML format.

4. **com.ibm.bcg.validation.OutboundValidationFactory** - This step validates the transformed target XML document from the previous transformation step.
5. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Custom XML Pass Through with Duplicate Check and Validation

### Purpose

This action is used with Custom XML documents. It can be used for documents coming from an external partner or coming from the internal partner. Duplicate ID checking is performed, and validation on the source document. This action is similar to the "Custom XML Pass Through with Duplicate Check" except there is the additional source document validation check performed.

### Configuration

- The source document XML Format needs the Duplicate Check Keys configured.
- Requires the validation map (XML SCHEMA) to be configured on either the source XML document.

### Modification

This action can be copied and modified. The steps that can be substituted with User defined steps is the ValidationFactory or additional User defined steps added.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.duplicate.ContentDuplicateProcessFactory** - Checks for a duplicate document based on the document ID. The XML Format for this source document must have the Duplicate ID configuration.
2. **com.ibm.bcg.validation.ValidationFactor**y - This step validates the source Custom XML document.
3. **com.ibm.bcg.passthrough.No_op** - Used to indicate the target document content type should not be derived from the document contents.
4. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Custom XML Pass Through with Duplicate Check

This action is used with custom XML documents. It can be used for documents coming from an external partner or coming from the internal partner. Duplicate ID checking is performed on the source document.

### Configuration

The source document XML Format needs the Duplicate Check Keys configured.

### Modification

This action cannot be copied into a new action, as the possible modification is adding a validation step, which is defined in the "Custom XML pass through with duplicate check and validation" action.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.duplicate.ContentDuplicateProcessFactory** - Checks for a duplicate document based on the document ID. The XML Format for this source document must have the Duplicate ID configuration.
2. **com.ibm.bcg.passthrough.No_op** - Used to indicate the target document content type should not be derived from the document contents.
3. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## Custom XML Pass Through with Validation

This action is used with custom XML documents that are coming from an external partner or coming from the internal partner. Validation is performed on the source document.

### Configuration

Requires the validation map (XML SCHEMA) to be configured on the source XML document.

### Modification

This action can be copied and modified. The ValidationFactory can be substituted with User defined step or additional User defined steps added.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.validation.ValidationFactory** - This step validates the source Custom XML document.
2. **com.ibm.bcg.passthrough.No_op** - Used to indicate the target document content type should not be derived from the document contents.
3. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## EDI De-envelope

This action is used with EDI Interchanges coming from an external partner. The EDI Interchange will be de-enveloped (the EDI Transactions extracted) these EDI Transactions will be reintroduced into WebSphere Partner Gateway for individual processing. The EDI Interchange document is not processed any further within WebSphere Partner Gateway.

### Configuration

Optional configuration in the Document Definitions.

### Modification

This action cannot be copied and modified

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.edi.business.process.EDIDenvFactory** – performs the EDI Interchange de-enveloping.
2. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## EDI Validate and EDI Translate

This action is used for EDI Transactions that were de-enveloped from an EDI Interchange by the EDI De-envelope action. These are from an external partner. The EDI Transaction documents will be validated and then transformed.

### Configuration

- Optional configuration in the Document Definitions
- Optional validation maps for the source EDI Transaction from the DIS Client or WTX design studio.
- Transformation maps from the DIS Client or WTX design studio.
- Participant Connection from any package / EDI - Any / Any to None / EDI - Any / Any should be setup with action defined as EDI De-envelope.

### Modification

This action can be copied and modified for adding additional User Exit steps.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.edi.business.process.EDISourceValidationFactory** – Validates the EDI Transaction. This step will also issue the EDI FA after processing all the EDI Transactions from the EDI Interchange.
2. **com.ibm.bcg.edi.business.process.EDITranslatorFactory** – Transforms the EDI Transaction into the target document.
3. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## XML Translate and EDI Validate
### Purpose

This action is used for Custom XML documents from the internal partner. The source XML document is transformed to an EDI transaction and validated. It is

then sent to either the backend or to an external partner. XML formats are used to identify the routing information.

### Configuration

- Optional configuration in the Document Definitions.
- Optional validation maps for the target EDI Transaction from the DIS Client.
- Transformation maps from the DIS Client or WDI design studio.

### Modification

This action can be copied and modified to remove the EDITargetValidationFactory or for adding additional User Exit steps.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.edi.business.process.XMLTranslatorFactory** – Transforms the source XML document to the target EDI Transaction.
2. **com.ibm.bcg.edi.business.process.EDITargetValidationFactory** – Validates the target EDI Transaction.
3. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## ROD (FlatFile) Translate and EDI Validate

This action is used for Record Oriented Documents (ROD/Flat File) from the internal partner. The source ROD document will be transformed into an EDI Transaction and validated.

### Configuration

- Optional configuration in the Document Definitions.
- Optional validation maps for the target EDI Transaction from the DIS Client.
- ROD Standard should be defined in the DIS client and compiled using a dummy transformation map.
- ROD splitter / Generic Document Processor should be added as per process handler in the receiver. This is to know the dictionary document and format.

### Modification

This action can be copied and modified to remove the EDITargetValidationFactory or for adding additional User Exit steps.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.edi.business.process.RODTranslatorFactory** – Transforms the source ROD document to the target EDI Transaction.
2. **com.ibm.bcg.edi.business.process.EDITargetValidationFactory** – Validates the target EDI Transaction.
3. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This

is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## ebMS Split and Parse

This action is for ebMS documents from an external partner. The payload attachments will be extracted and reintroduced into WebSphere Partner Gateway for individual processing. The ebMS document is not processed any further within WebSphere Partner Gateway.

### Configuration

No additional configuration is required.

### Modification

This action cannot be copied and modified.

### Steps

This action contains the following steps, which are run in sequence:

1. **com.ibm.bcg.server.EBMSSplitAndParse** – The payload attachments are extracted into individual documents.
2. **com.ibm.bcg.outbound.OutboundDocFactory** - Always required. Performs WebSphere Partner Gateway required processing on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## EDI interchange validate

EDI interchange validate is used during asynchronous integration with WTX. Individual transactions are extracted from the interchange by de-enveloping the interchange. Denvelope action will extract each transaction out of the interchange. Each transaction will produce a document that will be directly passed for validation

**Note:** The "Discard Envelope On Error" attribute cannot be used in the context of EDI interchange validation. If you try to set up the value for using this attribute, the value will be ignored.

### Configuration

- Participant Connection from <any package> / EDI – xxxx / XXX to None / EDI – xxxx / XXX should be setup with action defined as “EDI Interchange Validation”.
- Optionally, FA user can configure FA map.
- A channel for the functional Acknowledgement to flow through should be defined.

## WTX Transformation

EDI, XML ,and ROD or flat files are transformed using WTX.

The transformation of EDI using WTX can be asynchronous or synchronous. Synchronous is mostly used when a de-enveloped and validated transaction is sent to WTX for processing, but here the transaction would be enveloped back as it is

required for processing in WTX. Once the EDI transaction is successfully validated, it is passed to the WTX Transform EDI Transaction action. In asynchronous mode, EDI transactions are transformed in the backend, where WTX is deployed on WESB/WMB or WTX launcher.

The following points should be considered while using EDI as an input for transformation:

**Important:**

1. Always use a single character delimiter.
2. If you use combination of "/r/n" character delimiter, and if "/r" character delimiter is found at the segment delimiter position of Interchange Header, then the "/n" character delimiter will be ignored.
3. Modify the type tree accordingly.

### Configuration for synchronous

- Participant Connection from <any package> / EDI – xxxx / XXX to None / EDI – xxxx / XXX should be setup with action defined as EDI De-envelope.
- Participant Connection from <N/A> / XXXXXXX/ YYYYY to None / ZZZZZZ / BBBBBBB should be setup with action defined as “EDI Validate” & “WTX Transform EDI Transaction”.
- A WTX transformation map should be associated to this channel too.

### Configuration for asynchronous

- Participant Connection from <any package> / EDI – xxxx / XXX to None / EDI – xxxx / XXX should be setup with action defined as EDI De-envelope.
- Participant Connection from <N/A> / <edi version>/ transaction to <N/A> / <edi version> / transaction should be setup with action defined as EDI Validate.
- Participant Connection from <N/A> / <edi version> / transaction to <BI> / <edi interchange> / <ISA> / <UNB> / <UCS> should be setup with action defined as EDI Validate & EDI RE-ENVELOPE.

### Configuration for ROD and XML

- ROD transformation - participant Connection from <any package> / <any protocol (flat file) > / <any flat file> to <Any> / <ANY> / <Any> Format should be setup with action defined as “WTX Transform”.
- XML transformation - participant Connection from <any package> / <any protocol> / <any XML> to <Any> / <ANY> / <Any> Format should be setup with action defined as “WTX Transform”.

## WTX envelope

### Purpose

When we use WTX in asynchronous mode, it transforms and produces EDI transactions after WTX transform. This is sent to WebSphere partner Gateway for enveloping.

### Configuration

- Connection from <Backend> / <EDI Dictionary> / <EDI document> {EDI Trx} to <N/A> / <EDI X12/EDIFACT> / <EDI ISA/UNB> with action Pass Through Configure Enveloper profile in target end. (Channel-A).

- Connection from <NA> / <EDI Interchange> / <EDI ISA/UNB> to <ANY PACKAGE> / <EDI X12/EDIFACT> / <EDI ISA/UNB> with action as pass through. (Channel-B)
-

### EDI ReEnveloper

ReEnveloper is used to envelope individual transactions. It takes the enveloper headers from the source envelope and wraps each of the de-enveloped transactions with it.

#### Configuration

- A connection between source as transaction and Target as EDI Interchange with envelope profile set
- Set the action as EDI ReEnveloper

## SOAP Envelope Validate

The webservice request as a whole will be validated against SOAP1.1 schema as per industry standards. The action SOAP Envelope has the following steps, which are executed in sequence:

1. **com.ibm.bcg.validation.WebserviceFactory** – performs the Webservice request validation and returns WebserviceValidation handler.
2. **com.ibm.bcg.outbound.OutboundDocFactory** - always required. Performs processing required for WebSphere Partner Gateway on the target document. This is the last step and is automatically added by the Console to existing actions or newly created actions. This step does not appear in the configured handler list.

## SOAP Body validate

This feature validates SOAP Body or payload that is available under SOAP Envelope. Payload validation is supported only for XML payloads in SOAP Envelope. Industry standard schema location pointer under Payload XML is utilized for schema-based validation. Optionally, you can associate your schema with the concerned Webservice connection for validating the payload. The schema you have explicitly associated with Webservice connection will take precedence over the schema placed under payload XML. In the absence of a schema location pointer in payload XML, associate a schema under Webservice connection. Routing object attributes for both Webservice request and response are as follows:

- **ResponseValidation** – set the value of this attribute to “No” on the target side, if you do not want to validate a response document. The default value of this attribute is “Yes”.
- **ContentValidation** – this attribute allows you to enable or disable the content validation over payload XML. By default, content validation is enabled. If you set it to “No”, grammar validation will be performed.

The action SOAP Body contains the following steps, which are run in sequence: :

1. **com.ibm.bcg.validation.ValidationFactory** – performs the Webservice request validation.
2. **com.ibm.bcg.outbound.OutboundDocFactory** - always required. Performs processing required for WebSphere Partner Gateway on the target document. This is the last step and is automatically added by the Console to existing actions or the newly created actions. This step does not appear in the configured handler list.

To upgrade WebSphere Partner Gateway to include the feature of validating payload under SOAP envelope, see Administrator Guide.

## SOAP De-envelope

The SOAP Envelope must be de-enveloped and the SOAP Body element must be introduced for further processing. The routing object attributes to De-envelope SOAP Envelope is as follows:

- **De-Envelope SOAP Envelope** - supports only asynchronous communication. No SOAP fault or SOAP response is returned as it is one-Way Webservice basic profile support. In case of synchronous, De-envelope SOAP Envelope will fail the document and log error event.
- **Re-route De-enveloped Document** - this is a linked routing object attribute of **De-Envelope SOAP Envelope** action. If this routing object attribute is set to "Yes", the action **De-Envelope SOAP Envelope** has to introduce the extracted SOAP Body from the SOAP Envelope as a new document into WebSphere Partner Gateway. Additionally, the attachment must also be introduced as a new document. All the newly introduced documents will fall under the package N/A. To route them further, you need to configure N/A package based channel for the extracted payload and attached documents.
- **ConsumePayload** - this attribute is linked to **Re-route De-enveloped Document** attribute. It is used for suppressing the payload after extraction. If the value of this attribute and the value of **Re-route De-enveloped Document** is set to "Yes", then the payload is not extracted or routed from the SOAP envelope. The attachments alone are routed. In case this attribute is set to "No" and **Re-route De-enveloped Document** is set to "Yes", the Payload and attachments are routed separately. The default value of this attribute is "No".

The action De-Envelop SOAP Envelope contains the following steps, which are run in sequence: :

1. **com.ibm.bcg.validation.SOAPDeEnveloperFactory** – performs the Webservice request validation and returns SOAPDeEnveloper handler.
2. **com.ibm.bcg.outbound.OutboundDocFactory** - always required. Performs processing required for WebSphere Partner Gateway on the target document. This is the last step and is automatically added by the Console to existing actions or the newly created actions. This step does not appear in the configured handler list.

During instances wherein you want to de-envelope SOAP with attachment and route only attachments not the payload under SOAP body, the configuration is as follows:

- bcg.soap.ConsumePayload = Y (by default this value is N)
- bcg.soap.Re-RouteDe-EnvelopedDocument = Y (by default this value is Y)

When you want to de-envelope SOAP with attachment and route payload and attachments separately, the configuration is as follows:

- bcg.soap.ConsumePayload = N (by default this value is N)
- bcg.soap.Re-RouteDe-EnvelopedDocument = Y (by default this value is Y)

To upgrade WebSphere Partner Gateway to include the feature of validating payload under SOAP Envelope, see *WebSphere Partner Gateway Administrator Guide*.

## Modifying a user-defined action

### About this task

To configure a user-defined action, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Actions**.
2. Click the **View details** icon next to the name of the user-defined action you want to configure.

   The action, along with a list of handlers (action steps) already configured for that action, is listed.
3. Perform one or more of the following steps for each action you want to modify.
   a. Add a step by selecting the associated handler from the **Available List** and clicking **Add**. The handler is moved to the **Configured List**.
   b. Remove a handler by selecting the handler from the **Configured List** and clicking **Remove**. The handler is moved to the **Available List**.
   c. Rearrange the order in which the handlers are called by selecting the handler and clicking **Move Up** or **Move Down**.
   d. Cause a handler to be processed more than once by selecting it and then clicking **Repeat**.

      Remember that all handlers configured for an action are called and the steps that the handlers represent are performed in the order in which they appear in the **Configured List**.
   e. Configure the handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured is displayed.
4. Click **Save**.

## Creating actions

You can create an action in one of the following ways:

- Create a new action and associate handlers with the action.
- Copy a product-supplied action and, if necessary, modify the handlers associated with it.

### Creating a new action

#### About this task

To create a new action, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Actions**.
2. Click **Create**.
3. Enter a name for the action. This field is required.
4. Enter an optional description of the action.
5. Indicate whether the action is enabled for use.
6. For each step that will be invoked as part of the action, add the associated handler by selecting it from the **Available List** and clicking **Add**. The handler is moved to the **Configured List**.

   Remember that handlers are called by the action in the order in which they appear in the **Configured List**. Make sure you place the handlers in the correct order. You can use **Move Up** or **Move Down** to rearrange the order of the handlers or **Repeat** to cause a handler to be processed more than once.
7. Configure a handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured is displayed.
8. Click **Save**.

## Copying an action
### About this task

To create an action by copying an existing action, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Actions**.
2. From the Actions list, click the **Copy** icon next to the action you want to copy.
3. Enter a name for the action. This field is required.
4. Enter an optional description of the action.
5. Indicate whether the action is enabled for use.
6. Notice that one or more steps are already on the **Configured List**. These are the steps associated with the action you copied. For example, if you cloned the product-provided internal partner Cancellation of RosettaNet Process action, you would see the following list of available and configured handlers:

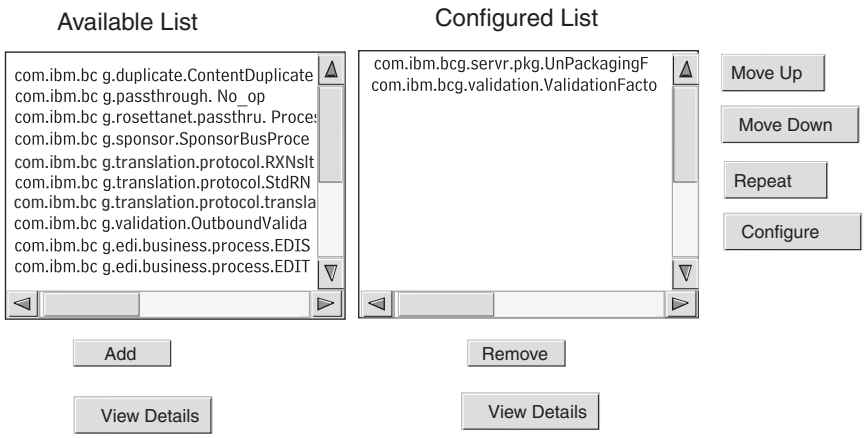



*Figure 19. Cloning an action*

   To modify the **Configured List**, perform one or more of the following steps:

   a. Add a step by selecting the associated handler from the **Available List** and clicking **Add**. The handler is moved to the **Configured List**.
   b. Remove a step by selecting the associated handler from the **Configured List** and clicking **Remove**. The handler is moved to the **Available List**.
   c. Rearrange the order in which the handlers are called by selecting the handler and clicking **Move Up** or **Move Down**.

      Remember that all handlers configured for an action are called and the steps associated with the handlers are performed in the order in which they appear in the **Configured List**.
   d. Configure the step by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured is displayed.

7. Click **Save.**

# Chapter 9. Configuring document types

This chapter describes how to configure the non-EDI documents you will be exchanging with your external partners and with your back-end applications. Configuring document types and interactions for EDI documents (with the exception of EDI documents that are being passed through) is described in Chapter 10, "Configuring EDI document flows," on page 161. Chapter 10, "Configuring EDI document flows," on page 161 also describes how to configure document types and interactions for XML and record-oriented-data (ROD) documents.

The chapter covers the following topics:



**Note:** You should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Overview of document types

A document definition is made up of, at minimum, a package, a protocol, and a document type. For some protocols, an activity, action, and signal can be specified. The document definitions specify the types of document that will be processed by WebSphere Partner Gateway.

Packaging refers to the logic that is required to package a document according to a specification, such as AS2. A protocol flow is the logic that is required to process a document that adheres to a certain protocol, such as EDI-X12. A document type describes what the document will look like.

The following sections briefly describe the overall steps for setting up a document type between the internal partner and a partner.

### Step 1: Make sure the document definition is available

#### About this task

Check to see whether a document definition exists (from the ones that are predefined with the system). If the flow does not already exist, you create it by uploading the necessary files, or by manually creating a custom definition.

As part of establishing the document definition, you can modify certain attributes. Attributes are used to perform various document-processing and routing functions,

such as validation, checking for encryption, and retry count. The attributes you set at the document definition level provide a global setting for the associated package, protocol, or document type. The attributes that are available vary, depending on the document definition. Attributes for EDI document definitions, for example, have attributes different from RosettaNet document definitions.

For example, if you specify a value for **Time to Acknowledge** on the AS package, it applies to all documents packaged with AS. (**Time to Acknowledge** specifies the amount of time to wait for an MDN (message disposition notification) acknowledgment before resending the original request.) If you later set the **Time to Acknowledge** attribute at the B2B capabilities level, that setting overrides the one set at the document definition level.

For attributes that can be set at all levels of the document definition, the values set at the document type level take precedence over those set at the protocol level, and the attributes set at the protocol level take precedence over the package level.

You must have the document type listed on the Manage document definitions page before you can create interactions. To manage document definition, see *Hub administration tasks Chapter of WebSphere Partner Gateway Administrator Guide.*

## Step 2: Create interactions

### About this task

Create interactions for the document types that have been defined. The interaction tells WebSphere Partner Gateway which actions to perform on a document. For some exchanges, you need only two flows, one to describe the document that is received into the hub (from the partner or internal partner) and one that describes the document that is sent from the hub (to the external partner or internal partner). However, if the hub is sending or receiving an EDI interchange that will be split into individual transactions or in which acknowledgments are required, you will actually create multiple interactions to perform the exchange. To manage interactions, see*Hub administration tasks Chapter of WebSphere Partner Gateway Administrator Guide.*

## Step 3: Create partner profiles, destinations, and B2B capabilities

### About this task

Create partner profiles for the internal partner and for external partners. Define destinations (which determine where documents will be sent) and B2B capabilities, which specify the documents the internal partner and external partners are capable of sending and receiving. The B2B capabilities page lists all the document types that have been defined.

You can set attributes at the B2B capabilities level. Any attributes you set at this level override those set at the document definition level. For example, if you set **Time to Acknowledge** to 30 at the document definition level for AS package but then set it to 60 at the B2B capabilities level, the value of 60 is used. Setting an attribute at the B2B level lets you tailor the attribute to a specific partner.

## Step 4: Activate connections

### About this task

Activate connections between the internal partners and external partners. The connections that are available are based on the created interactions. The interacts are based on B2B capabilities. The interactions depend on the document definitions being available.

For some exchanges, only one connection is required. For example, if a partner is sending a binary document to an internal partner back-end application, only one connection is needed. For the exchange of EDI interchanges in which the interchange is de-enveloped and the individual transactions are transformed, however, multiple connections are set up.

**Note:** For EDI interchanges that are being passed through as is, only one connection is required.

You can set attributes at the connection level. Any attributes you set at this level override those set at the B2B capabilities level. For example, if you set **Time to Acknowledge** to 60 for the AS2 package at the B2B capabilities level but then set it to 120, the value of 120 is used. Setting a value for an attribute at the connection level lets you further tailor the attribute, depending on the routing requirements of the partners and applications involved.

## An example flow

### About this task

By default, several packaging methods are enabled. To illustrate the overall procedure for establishing document definitions, consider the case in which you have an agreement with an external partner to receive an EDI interchange that adheres to the EDI-X12 standard. The partner will send the document in AS2 packaging. You will specify that the interchange be sent as is (without transformation) to a back-end application with no packaging.

1. At the Manage Document Definitions page, verify that the document definition (which describes the type of document that will flow into the hub from the partner) is enabled.
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Click the **Expand** icon next to **Package: AS**. Notice that **EDI-X12** is already listed.
   c. Click the **Expand** icon next to **Protocol: EDI-X12**. Notice that **Document Type: ISA** is already listed.
2. With the Manage Document Definition page still displayed, verify the second document definition (which describes the type of document that will flow to the back-end application) is enabled.
   a. Click the **Expand** icon next to **Package: None**. Notice that **EDI-X12** is already listed.
   b. Click the **Expand** icon next to **Protocol: EDI-X12**. Notice that **Document Type: ISA** is already listed.
3. Create an interaction that describes whether the document type will be a source type or a receiver type.
   a. With the Manage Document Definition page still displayed, click **Manage Interactions** link.

b. In the Source column, expand **Package: AS**, **Protocol: EDI-X12 (ALL)**, and then click **Document Type: ISA** so that the radio button is selected.
c. In the Target column, expand **Package: None**, **Protocol: EDI-X12 (ALL)**, and then click **Document Type: ISA** so that the radio button is selected.
d. In this example, no transformation is occurring. Therefore, do not select anything from the **Transformation map** list.
e. From the **Action list**, select **Pass Through**.
f. Click **Save**.

At this point, you have specified that the hub is capable of accepting EDI-X12 interchanges (ISA standard) packaged as AS. You have also specified that the hub is capable of sending EDI-X12 interchanges (ISA standard) with no packaging. Further, you have specified that no transformation is to occur with the interchange; it is simply passed through to the back-end application (after the AS headers are removed).

*Table 5. Product-provided interactions for Open PGP*

| On the sender side, set this connection | On the receiver side, set this connection |
|---|---|
| None/EDI-X12/ISA to None/EDI-X12/ISA | None/EDI-X12/ISA to None/EDI-X12/ISA |
| Backend integration/EDI-X12/ISA to None/EDI-X12/ISA | None/EDI-X12/ISA to None/EDI-X12/ISA |

**Note:** EDI-X12 is not present in Backend Integration by default, so add "EDI-X12" to the Backend Integration context. After adding "EDI-X12" to the Backend Integration context, add "ISA" to Backend Integration - EDI-X12 context.

You have not yet specified which partner is capable of sending this type of interchange to the hub. You define that when you set up the partner profile and the partner's B2B capabilities. (You also define a profile and B2B capabilities for the internal partner back-end system.) After you perform these tasks, you then create a connection between the partner and the back-end application. Figure 20 shows the connection between the partner and the internal partner back-end application for this example.

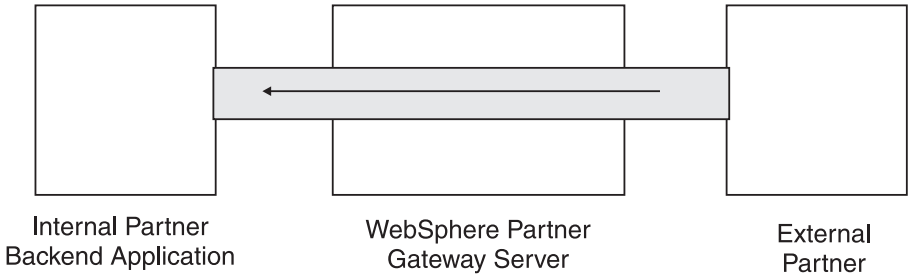



*Figure 20. A one-way connection from a partner to the internal partner*

You verify that a connection exists using the Manage Connections page (**Account Admin > Connections > Partner Connections**). On the Manage Connections page, you select the partner from the **Source** list, internal partner from the **Target** list, and click **Search**. The one available connection is listed. If necessary, you can modify attributes and actions, as will be described in subsequent sections.

There are three types of document definitions--ones supplied with the system that you can select from the console, ones that are already defined but not yet on the

Community Console (you upload these definitions either from the WebSphere Partner Gateway installation medium or from another location), and those that you create yourself. For each type of document definition, you can (or sometimes must) specify attributes or upload maps that further define the document type.

## Binary documents

Binary documents are the documents, which are passed through the hub as is. These documents are exchanged between an external partner and an internal partner by using a back-end application. You must have the profiles and business-to-business (B2B) capabilities of the internal partners and external partners defined before you can create connections between them. If the default internal partner is not used, then the receiverID of the internal partner must be explicitly set. When the binary document is routed through HTTP transport by using basic authentication, the receiverID can be passed through **X-aux-receiver-id** attribute. Using FTP protocol, an external partner can send binary documents to the hub. The binary protocol is already available for the AS, None, and Backend Integration packages; therefore, "Step 1: Make sure the document definition is available" on page 99 is already done.

**Note:** You can add attributes at Package, Protocol, or Document Type level to modify default processing by clicking the **Edit Attribute Values** icon. By default, no attributes are associated with the binary protocol or document type.

By default, four interactions involving binary documents are already provided for WebSphere Partner Gateway and three interactions are newly provided for Open PGP. For these interactions, it is not necessary for you to perform "Step 2: Create interactions" on page 100. Interactions are supplied for the following exchanges:

*Table 6. Product-provided interactions*

| Source Package/Protocol/Document Type | Target Package/Protocol/Document Type |
|---|---|
| AS/Binary/Binary | Backend Integration/Binary/Binary |
| Backend Integration/Binary/Binary | AS/Binary/Binary |
| AS/Binary/Binary | None/Binary/Binary |
| None/Binary/Binary | AS/Binary/Binary |

For OpenPGP, manually enable the following supported interactions from the console of WebSphere Partner Gateway:

*Table 7. OpenPGP supported interactions*

| Source Package/Protocol/Document Type | Target Package/Protocol/Document Type |
|---|---|
| None/Binary/Binary | None/Binary/Binary |
| Backend Integration/Binary/Binary | None/Binary/Binary |

For the exchange of binary documents, you still have to perform:

- "Step 3: Create partner profiles, destinations, and B2B capabilities" on page 100, which is described in Chapter 3, "Creating and setting up partners," on page 23, and Chapter 11, "Creating destinations," on page 209.
- "Step 4: Activate connections" on page 101, which is described in Chapter 12, "Managing connections," on page 231.

## EDI documents with Pass Through action

WebSphere Partner Gateway provides the capability to de-envelope and transform EDI interchanges, a process described in Chapter 10, “Configuring EDI document flows,” on page 161.

Figure 21 shows the flow of an EDI interchange that is being passed through from a partner to the internal partner.

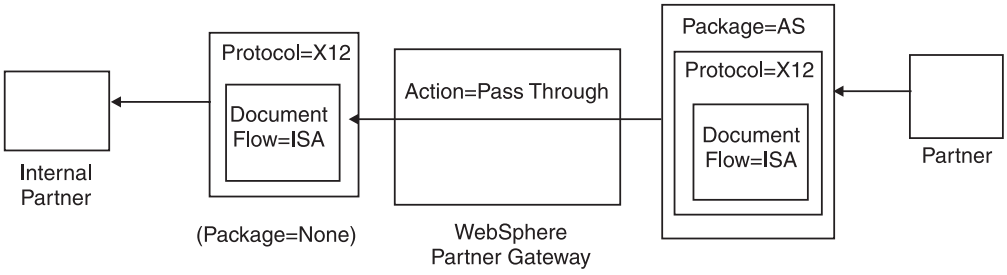



*Figure 21. Incoming EDI interchange with Pass Through action*

In this example, the AS2 headers are removed, but otherwise the interchange is left intact and flows through the system to the destination of the internal partner.

In synchronous transformation of EDI transaction using WTX (EDI to Any), if the transformation has more than one output, then based on the reroute flag the children will be passed directly to the outbound workflow or rerouted into the fixed inbound workflow for it to pass through a new channel. In Asynchronous case, the WTX will send EDI transactions to WPG for enveloping. You need to set connections for the two channels - <none> / <EDI Dictionary> / <EDI document > {EDI Trx} with Pass Through and <NA> / <EDI interchange> / <,EDI ISA / UNB> to <Any Package> / <EDI X12 / <FACT> / <EDI ISA / UNB with action as Pass Through.

## Creating document definitions

### About this task

The document type for EDI passthrough exchanges is already provided (by default) on the Manage Document Definitions page, as described in “An example flow” on page 101. If you want to modify any of the attributes that have default values or set an attribute that has no assigned value, you can use the Manage Document Definitions page to perform this task.

For example, suppose you want to change the **Time to Acknowledge** attribute for an EDI document packaged with AS. These are the steps you take:

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click the **Edit Attribute Values** icon next to **Package: AS**.
3. Scroll down to the section of the page titled **Document Definition Context Attributes**.
4. In the **Time to Acknowledge** row, type a different value in the **Update** column.
5. Click **Save**.

Note that you changed a package attribute in this example. The attributes for protocol (for example, EDI-X12) and document type (for example, ISA) are not

relevant for a Pass Through action. This package attribute applies to all documents wrapped in AS packaging.

## Creating interactions

### About this task

To create the interaction for an EDI interchange with Pass Through action, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Under **Source**, expand **Package: AS** and **Protocol: EDI-X12** and then select **Document Type: ISA**.
4. Under **Target**, expand **Package: None** and **Protocol: EDI-X12** and then select **Document Type: ISA**.
5. Optionally, select a **Transformation Map**.
6. From the **Action** list, select **Pass Through**.

Steps 1 through 4 have enabled WebSphere Partner Gateway to accept an EDI-X12 interchange packaged as AS from a source partner, to send an EDI-X12 interchange with no packaging to the target partner, and to have the interchange pass through from the source to the target.

If you want to set up an interaction that has the source document packaged as None/EDI-X12/ISA and the target document packaged as AS/EDI-X12/ISA, expand **Package: None** in step 3 (in the **Source** column) and expand **Package: AS** in step 4 (in the **Target** column).

# RosettaNet documents

RosettaNet is an organization that provides open standards to support the exchange of business messages between partners. For more information about RosettaNet, see http://www.rosettanet.org. The standards include RosettaNet Implementation Framework (RNIF) and Partner Interface Process (PIP) specifications. RNIF defines how partners exchange messages by providing a framework of message packaging, transfer protocols, and security. There are two released versions: 1.1 and 2.0. A PIP defines a public business process and the XML-based message formats to support the process.

WebSphere Partner Gateway supports RosettaNet messaging using RNIF 1.1 and 2.0. When the hub receives a PIP message, it validates and transforms the message to send it to the appropriate back-end system. WebSphere Partner Gateway provides a protocol for packaging the transformed message into a RosettaNet Service Content (RNSC) message that the back-end system can handle. See the *WebSphere Partner Gateway Enterprise Integration Guide* for information about the packaging used on these messages to provide routing information.

The hub can also receive RNSC messages from back-end systems and create the appropriate PIP message and send the message to the appropriate trading partner (a partner). You provide the document definitions for the RNIF version and the PIPs you want to use.

In addition to providing routing capability for RosettaNet messages, WebSphere Partner Gateway maintains a state for each message it handles. This enables it to

resend any messages that fail until the number of attempts reaches a specified threshold. The Event Notification mechanism alerts back-end systems if a PIP message cannot be delivered. Additionally, the hub can automatically generate 0A1 PIPs to send to appropriate partners if it receives certain Event Notification messages from back-end systems. See the *WebSphere Partner Gateway Enterprise Integration Guide* for more information about Event Notification.

## RNIF and PIP document type packages

To support RosettaNet messaging, WebSphere Partner Gateway provides two sets of zipped files called packages. The *RNIF packages* consist of document definitions required to support the RNIF protocol. These packages are in the B2BIntegrate directory.

For RNIF V1.1, the packages are:

- Package_RNIF_1.1.zip
- Package_RNSC_1.0_RNIF_1.1.zip

For RNIF V02.00, the packages are:

- Package_RNIF_V02.00.zip
- Package_RNSC_1.0_RNIF_V02.00.zip

The first package in each pair provides the document definitions required to support RosettaNet communications with partners, and the second package provides the document definitions required to support RosettaNet communications with back-end systems.

The second set of packages consists of PIP document type packages. Each PIP document type package has a Packages directory containing an XML file and a GuidelineMaps directory containing XSD files. The XML file specifies the document definitions that define how WebSphere Partner Gateway handles the PIP and define the exchanged messages and signals. The XSD files specify the format of the PIP messages and define acceptable values for XML elements in the messages. The zipped files for 0A1 PIPs also have an XML file that the hub uses as a template to create 0A1 documents.

The PIPs for which WebSphere Partner Gateway provides PIP document type packages are:

- PIP 0A1 Notification of Failure v1.0
- PIP 0A1 Notification of Failure V02.00.00
- PIP 2A1 Distribute New Product Information V02.00.00
- PIP 2A12 Distribute Product Master V01.03.00
- PIP 3A1 Request Quote V02.00.00
- PIP 3A2 Request Price and Availability R02.01.00
- PIP 3A4 Request Purchase Order V02.02.00
- PIP 3A4 Request Purchase Order V02.00
- PIP 3A5 Query Order Status R02.00.00
- PIP 3A6 Distribute Order Status V02.02.00
- PIP 3A7 Notify of Purchase OrderUpdate V02.02.00
- PIP 3A8 Request Purchase Order Change V01.02.00
- PIP 3A8 Request Purchase Order Change V01.03.00
- PIP 3A9 Request Purchase Order Cancellation V01.01.00

- PIP 3B2 Notify of Advance Shipment V01.01.00
- PIP 3B3 Distribute Shipment Status R01.00.00
- PIP 3B11 Notify of Shipping Order R01.00.00A
- PIP 3B12 Request Shipping Order V01.01.00
- PIP 3B13 Notify of Shipping Order Confirmation V01.01.00
- PIP 3B14 Request Shipping Order Cancellation V01.00.00
- PIP 3B18 Notify of Shipping Documentation V01.00.00
- PIP 3C1 Return Product V01.00.00
- PIP 3C3 Notify of Invoice V01.01.00
- PIP 3C4 Notify of Invoice Reject V01.00.00
- PIP 3C6 Notify of Remittance Advice V01.00.00
- PIP 3C7 Notify of Self-Billing Invoice V01.00.00
- PIP 3D8 Distribute Work in Process V01.00.00
- PIP 4A1 Notify of Strategic Forecast V02.00.00
- PIP 4A3 Notify of Threshold Release Forecast V02.00.00
- PIP 4A4 Notify of Planning Release Forecast R02.00.00A
- PIP 4A5 Notify of Forecast Reply V02.00.00
- PIP 4B2 Notify of Shipment Receipt V01.00.00
- PIP 4B3 Notify of Consumption V01.00.00
- PIP 4C1 Distribute Inventory Report V02.03.00
- PIP 4C1 Distribute Inventory Report V02.01
- PIP 5C1 Distribute Product List V01.00.00
- PIP 5C2 Request Design Registration V01.00.00
- PIP 5C4 Distribute Registration Status V01.02.00
- PIP 5D1 Request Ship From Stock And Debit Authorization V01.00.00
- PIP 6C1 Query Service Entitlement V01.00.00
- PIP 6C2 Request Warranty Claim V01.00.00
- PIP 7B1 Distribute Work in Process V01.00.00
- PIP 7B5 Notify of Manufacturing Work Order V01.00.00
- PIP 7B6 Notify of Manufacturing Work Order Reply V01.00.00

For each PIP, there are four PIP document type packages:

- For RNIF 1.1 messaging with partners
- For RNIF 1.1 messaging with back-end systems
- For RNIF 2.0 messaging with partners
- For RNIF 2.0 messaging with back-end systems

Each PIP document type package follows a specific naming convention you can use to identify whether the package is for messages between WebSphere Partner Gateway and partners or between WebSphere Partner Gateway and back-end systems. The naming convention also identifies the RNIF version, PIP, and PIP version that the package supports. For PIP document type packages used for messaging between WebSphere Partner Gateway and partners, the format is:

`BCG_Package_RNIF<RNIF_version>_<PIP><PIP_version>.zip`

For PIP document type packages used for messaging between WebSphere Partner Gateway and back-end systems, the format is:

```
BCG_Package_RNSC<Backend_Integration_version>_RNIF<RNIF_version>_
<PIP><PIP_version>.zip
```

For example, the BCG_Package_RNIF1.1_3A4V02.02.zip is for validating documents for Version 02.02 of the 3A4 PIP sent between partners and WebSphere Partner Gateway using the RNIF 1.1 protocol. For PIP document type packages for communicating with back-end systems, the name of the package must also identify the protocol used to send the RosettaNet contents to the back-end systems. See the *WebSphere Partner Gateway Enterprise Integration Guide* for information about the packaging used for these messages.

## Creating document definitions

### About this task

For RosettaNet messaging, WebSphere Partner Gateway requires the RNIF packages for the version of RNIF used to send the messages. For each PIP that WebSphere Partner Gateway supports, it requires the two PIP document type packages for the RNIF version. For example, to support the 3A4 PIP over RNIF 2.0, WebSphere Partner Gateway requires the following packages:

- Package_RNIF_V02.00.zip
- Package_RNSC_1.0_RNIF_V02.00.zip
- BCG_Package_RNIFV02.00_3A4V02.02.zip
- BCG_Package_RNSC1.0_RNIFV02.00_3A4V02.02.zip

The first package supports RosettaNet messaging with partners and the second package supports RosettaNet messaging with back-end systems. The third and fourth packages enable WebSphere Partner Gateway to pass 3A4 messages between partners and back-end systems using RNIF 2.0.

To upload the RosettaNet packages:

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Upload/Download Packages**.
3. Select **No** for **WSDL Package**.
4. Click **Browse** and select the RNIF package for communicating with partners.

   The RNIF packages are located, by default, in the B2BIntegrate/Rosettanet directory on the installation medium. For example, if you were uploading RNIF Version 2.00 package, you would browse to the B2BIntegrate/Rosettanet directory and select: Package_RNIF_V0200.zip.
5. Make sure **Commit to Database** is set to **Yes**.
6. Click **Upload**.
7. Click **Browse** again and select the RNIF package for communicating with back-end applications.

   For example, if you were uploading the RNIF Version 2.00 package, you would browse to the B2BIntegrate/Rosettanet directory and select Package_RNSC_1.0_RNIF_V02.00.zip.
8. Click **Upload**.

   The packages needed to communicate with partners or with the back-end system are now installed in the system. If you check the Manage Document Definitions page, you see an entry for **Package: RNIF/Protocol: RosettaNet**, which represents the packaging for communicating with partners, and **Package: Backend Integration/Protocol: RNSC**, which represents the packaging for communicating with back-end applications.

9. For each PIP you want to support, upload the PIP document type package for the PIP and for the RNIF version you are supporting. For example, to upload the 3A6 PIP (Notify of Remittance Advice) to be sent to a partner, perform the following steps:
   a. Click **Browse** and select BCG_Package_RNIFV02.00_3C6V02.02 from the B2BIntegrate/Rosettanet directory.
   b. Make sure **Commit to Database** is set to **Yes**.
   c. Click **Upload**.

   The 3C6V02.02 PIP now appears as the document type underneath **Package: RNIF/Protocol: RosettaNet** on the Manage Document Definitions page. An activity, action, and two signals are also displayed. They are included in the upload of the PIP.

   To upload the 3A6 PIP to be sent to the back-end application, perform the following steps:
   a. Click **Browse** and select BCG_Package_RNSC1.0_RNIFV02.00_3C6V02.02.zip.
   b. Make sure **Commit to Database** is set to **Yes**.
   c. Click **Upload**.

   The 3C6V02.02 PIP now appears as the document type underneath **Package: Backend Integration/Protocol: RNSC** on the Manage Document Definitions page. If WebSphere Partner Gateway does not provide a package for the PIP or PIP version you want to use, you can create your own and upload it. See "Creating PIP document definition packages" on page 353for more information.

# Configuring attribute values

## About this task

For PIP document definitions, most of the values of the attributes are already set and do not need to be configured. However, you do need to set the following attributes:

RNIF (1.0) package

- **GlobalSupplyChainCode** - Identify the type of supply chain used by the partner. The types are Electronic Components, Information Technology, and Semiconductor manufacturing. This attribute does not have a default value.

RNIF (V02.00) package

- **Encryption** - Set whether the PIPs must have an encrypted payload, an encrypted container and payload, or no encryption. The default value is None.
- **Sync Ack Required** - Set to yes if the partner wants to receive the receipt acknowledgment. Set to No if a 200 is requested.
- **Sync Supported** - Set whether the PIP supports synchronous message exchanges. The default value is No.

Note that the PIPs for which WebSphere Partner Gateway provides PIP document type packages are not synchronous. As a result, you do not need to change the Sync Ack Required and Sync Supported attributes for these PIPs.

**Note:** The behavior of the Sync Ack Required attribute differs between 1-way and 2-way PIPs. For a 2-way PIP, when Sync Ack Required is set to No, this setting takes precedence over a NonRep of Rec setting of Yes. For example, suppose you send a 3A7 with the following settings:

- SiqReq=Y
- NonRepofRec=Y
- SyncSupported=Y
- SyncAckReq=N

For a 2-way PIP, you receive an error message on the incoming document. On a 1-way PIP, however, you see the incoming document on the console, and a 0KB 200 is returned to the partner.

To set the attributes, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Expand** icons to individually expand a node to the appropriate document definition level or select **All** to expand all displayed document definition nodes.
3. In the **Actions** column, click the **Edit Attribute Values** icon for the package (for example, Package: RNIF (1.1) or Package: RNIF (V02.00)) you want to edit.
4. In the **Document Definition Context Attributes** section, go to the **Update** column of the attribute you want to set and select or type the new value. Repeat for each attribute that you want to set.
5. Click **Save**.

**Note:** You can also update RosettaNet attributes at the connection level by clicking **Attributes** for the source or target and then entering or changing the values in the **Update** column. Refer to "Specifying or changing attributes" on page 232.

## Creating interactions

### About this task

The following process describes how to create an interaction between a back end system and a partner. Note that you must create an interaction for each PIP that you want to send and one for each PIP that you want to receive.

Before you begin, make sure that the appropriate RNIF document definitions have been uploaded and that the packages for the PIP you want to use have been uploaded. If you want the ability to generate an 0A1 PIP (Notification of Failure), make sure you have uploaded that PIP, as described in step 9 on page 109.

To create an interaction for a particular PIP, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand the **Source** tree to the **Action** level and expand the **Target** tree to the **Action** level.
4. In the trees, select the document definitions to use for the source context and the target context. For example, if the partner is the initiator of a 3C6 PIP (a one-action PIP), select the following document definitions:

*Table 8. 3C6 PIP initiated by a partner*

| Source | Target |
|---|---|
| Package: RNIF (V02.00) | Package: Backend Integration (1.0) |
| Protocol: RosettaNet (V02.00) | Protocol: RNSC (1.0) |

*Table 8. 3C6 PIP initiated by a partner (continued)*

| Source | Target |
|---|---|
| Document Type: 3C6 (V01.00) | Document Type: 3C6 (V01.00) |
| Activity: Notify of Remittance Advice | Activity: Notify of Remittance Advice |
| Action: Remittance Advice Notification Action | Action: Remittance Advice Notification Action |

If the back end system is the initiator of the 3C6 PIP, select the following document definitions:

*Table 9. 3C6 PIP initiated by a back end system*

| Source | Target |
|---|---|
| Package: Backend Integration (1.0) | Package: RNIF (V02.00) |
| Protocol: RNSC (1.0) | Protocol: RosettaNet (V02.00) |
| Document Type: 3C6 (V01.00) | Document Type: 3C6 (V01.00) |
| Activity: Notify of Remittance Advice | Activity: Notify of Remittance Advice |
| Action: Remittance Advice Notification Action | Action: Remittance Advice Notification Action |

For a two-action PIP such as 3A4 initiated by a partner, select the following document definitions for the first action:

*Table 10. 3A4 PIP initiated by a partner*

| Source | Target |
|---|---|
| Package: RNIF (V02.00) | Package: Backend Integration (1.0) |
| Protocol: RosettaNet (V02.00) | Protocol: RNSC (1.0) |
| Document Type: 3A4 (V02.02) | Document Type: 3A4 (V02.02) |
| Activity: Request Purchase Order | Activity: Request Purchase Order |
| Action: Purchase Order Request Action | Action: Purchase Order Request Action |

If a back-end system initiates the two-action 3A4 PIP, select the following document definitions for the first action:

*Table 11. 3A4 PIP initiated by a back-end system*

| Source | Target |
|---|---|
| Package: Backend Integration (1.0) | Package: RNIF (V02.00) |
| Protocol: RNSC (1.0) | Protocol: RosettaNet (V02.00) |
| Document Type: 3A4 (V02.02) | Document Type: 3A4 (V02.02) |
| Activity: Request Purchase Order | Activity: Request Purchase Order |
| Action: Purchase Order Request Action | Action: Purchase Order Request Action |

5. In the Action field, select **Bi-Directional Translation of RosettaNet and RosettaNet Service Content with Validation**.
6. Click **Save**.
7. If you are setting up a two-action PIP, repeat the steps needed to create the interaction for the second action. For example, select the following document definitions for the second action for a 3A4 PIP initiated by a partner. This is the action in which the back-end system sends the response.

*Table 12. 3A4 PIP initiated by a partner (second action)*

| Source | Target |
|---|---|
| Package: Backend Integration (1.0) | Package: RNIF (V02.00) |
| Protocol: RNSC (1.0) | Protocol: RosettaNet (V02.00) |
| Document Type: 3A4 (V02.02) | Document Type: 3A4 (V02.02) |
| Activity: Request Purchase Order | Activity: Request Purchase Order |
| Action: Purchase Order Confirmation Action | Action: Purchase Order Confirmation Action |

For the second action for a back-end system initiated 3A4 PIP, select the following document definitions:

*Table 13. 3A4 PIP initiated by a back-end system (second action)*

| Source | Target |
|---|---|
| Package: RNIF (V02.00) | Package: Backend Integration (1.0) |
| Protocol: RosettaNet (V02.00) | Protocol: RNSC (1.0) |
| Document Type: 3A4 (V02.02) | Document Type: 3A4 (V02.02) |
| Activity: Request Purchase Order | Activity: Request Purchase Order |
| Action: Purchase Order Confirmation Action | Action: Purchase Order Confirmation Action |

8. If you want to generate the 0A1 Notification of Failure, create an interaction for XMLEvent.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link** .
   c. Expand the **Source** tree to the **Document Type** level and expand the **Target** tree to the **Document Type** level.
   d. Select the following document definitions:

*Table 14. XML Event document definition*

| Source | Target |
|---|---|
| Package: Backend Integration (1.0) | Package: Backend Integration (1.0) |
| Protocol: XMLEvent (1.0) | Protocol: XMLEvent (1.0) |
| Document Type: XMLEvent (1.0) | Document Type: XMLEvent (1.0) |

   e. In the Action field, select **Pass Through**.
   f. Click **Save**.
9. Create an interaction for XMLEvent to 0A1 RNSC.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Expand the **Source** tree to the **Document Type** level and expand the **Target** tree to the **Activity** level.
   d. Select the following document definitions:

*Table 15. XML Event to 0A1 document definition*

| Source | Target |
|---|---|
| Package: Backend Integration (1.0) | Package: Backend Integration (1.0) |
| Protocol: XMLEvent (1.0) | Protocol: RNSC (1.0) |

*Table 15. XML Event to 0A1 document definition (continued)*

| Source | Target |
|---|---|
| Document Type: XMLEvent (1.0) | Document Type: 0A1 (V02.00) |
| | Activity: Distribute Notification of Failure. |

e. In the Action field, select **Bi-directional Translation of RosettaNet and XML with Validation**.

f. Click **Save**.

**Note:** For enabling or disabling XMLEvents, see *Enabling or disabling XMLEvents section of Enterprise Integration Guide*

# Viewing RosettaNet documents

## About this task

The RosettaNet Viewer displays information about RosettaNet documents. You can display raw documents and associated document processing details and events using specific search criteria. This information is useful if you are trying to determine whether a document was successfully delivered or to determine the cause of a problem.

To display the RosettaNet Viewer complete the following:

1. Click **Viewers > RosettaNet Viewer**.
2. Select the appropriate search criteria from the lists, as described in Table 16.

*Table 16. RosettaNet search criteria*

| Value | Description |
|---|---|
| Start Date and Time | The date and time of the process that was initiated. |
| End Date and Time | The date and time of the process that was completed. |
| Source and Target Partner | Identifies the source (initiating) and the target (receiving) partners (internal partner only). |
| Partner | Indicates whether the search applies to all partners or internal partner only. |
| My role is the | Indicates whether the search looks for documents in which the partner is either Target or Source. |
| Source Business ID | Business identification number of initiating partner, for example, DUNS. |
| Operation Mode | Production or test. Test is available only on systems that support the test operation mode. |
| Protocol | Protocols available to the partners. |
| Document Type | The specific business process. |
| Process Instance ID | Unique identification number assigned to the process. Criteria can include asterisk (*) wildcard. |
| Sort by | Sort results by:<br>• Target Timestamp<br>• Document Type<br><br>The default is Target Timestamp. |
| Descend or Ascend | Descend displays the most recent time stamp or the beginning of the first alphabet.<br><br>Ascend displays either the oldest time stamp or the end of the first alphabet.<br><br>The default is Descend. |

*Table 16. RosettaNet search criteria (continued)*

| Value | Description |
|---|---|
| Results Per Page | Specifies the number of results displayed per page. |

3. Click **Search**.

## CIDX documents

CIDX is a robust trade association and standards body whose mission is to improve the ease, speed and cost of conducting business electronically between chemical companies and their trading partners. CIDX has various initiatives which drive the standards for chemical industry. CIDX's Chem eStandards initiative is of interest in this document. Chem eStandard are the uniform standards of data exchange developed specifically for the buying, selling and delivery of chemicals. Chem eStandards consists of following:

- ChemXML or the Chem eStandards message specifications: v2.0, v2.0.1, v2.0.2, v3.0 and v4.0.
- Chem eStandards envelope and security specification: v2.0 and v3.0.

For packaging CIDX always uses RNIF 1.1. It is important to note that RNIF 1.1 is always asynchronous. Therefore CIDX document exchanges are always asynchronous.

CIDX consists of packaging and transactions whereas RosettaNet consists of packaging and PIPs (partner interchange processes). CIDX uses RNIF 1.1 packaging. Transactions are as defined by ChemXML standard. Each version of ChemXML standard defines transactions. All transactions of ChemXML under given ChemXML standard version have same version as that of ChemXML standard. Unlike RosettaNet CIDX does not require conformance to process definition. CIDX is more concerned with structure of the transaction and exchanging messages in secure manner.

To continue the comparison, RosettaNet is the administering authority for RosettaNet standard similarly CIDX is the administering authority of CIDX standard. RosettaNet defines RNIF packaging and PIPs. RosettaNet messages can use RNIF 1.1 or RNIF 2.0. RosettaNet defined PIPs give the message set and process choreography. CIDX always uses RNIF 1.1 as defined by RosettaNet. Since CIDX is the administering body, the RNIF envelope needs to be constructed as defined by Chem eStandards envelope and security specification. This specification is based on RosettaNet implementation. CIDX does NOT uses PIPs defined by RosettaNet. Instead CIDX uses Chem eStandards message specification.

For more information about CIDX, see http://www.cidx.org. CIDX standards can be downloaded from here: http://www.cidx.org. Chem eStandards Envelope and Security Version 3.0 can be found at http://www.cidx.org/Portals/0/Publications/Envelope_and_Security_v3.0.pdf.

WebSphere Partner Gateway supports the following Chem eStandards:

- Chem eStandards envelope and security specification v3.0.
- ChemXML or the Chem eStandards message specifications v4.0.

## RNIF and PIP document type packages for CIDX

CIDX uses RNIF1.1. To support CIDX, WebSphere Partner Gateway provides two sets of zipped files called packages. The *RNIF packages* consist of document definitions required to support the RNIF protocol. These packages are in the B2BIntegrate directory.

For RNIF V1.1, the packages are:

- Package_RNIF_1.1.zip
- Package_RNSC_1.0_RNIF_1.1.zip

The first package provides the document definitions required to support CIDX communications with partners, and the second package provides the document definitions required to support CIDX communications with back-end systems.

The second set of packages consists of PIP document type packages. Each PIP document type package has a Packages directory containing an XML file and a GuidelineMaps directory containing XSD files. The XML file specifies the document definitions that define how WebSphere Partner Gateway handles the PIP and define the exchanged messages and signals. The XSD files specify the format of the PIP messages and define acceptable values for XML elements in the messages. The zipped files for 0A1 PIPs also have an XML file that the hub uses as a template to create 0A1 documents.

For CIDX, WebSphere Partner Gateway provides document type packages for E41 ChemXML Version 4.0 Order Create and E42 ChemXML Version 4.0 Order Response.

The naming convention of the supplied CIDX packages is the same as the packages supplied for RosettaNet. For example, the BCG_Package_RNIF1.1_E414.0.zip is for validating documents for v4.0 for the E41 PIP send between partners and WPG using RNIF1.1.

## Creating document definitions

### About this task

For CIDX messaging, WebSphere Partner Gateway requires the RNIF packages for the version of RNIF used to send the messages. For each PIP that WebSphere Partner Gateway supports, it requires the two PIP document type packages for the RNIF version. For example, to support the E41 PIP over RNIF1.1, WebSphere Partner Gateway requires the following packages:

- Package_RNIF_1.1.zip
- Package_RNSC_1.0_RNIF_1.1.zip
- BCG_Package_RNIF1.1_E414.0.zip
- BCG_Package_RNSC1.0RNIF1.1_E414.0.zip

The first package supports CIDX messaging with partners, and the second package supports CIDX messaging with back-end systems. The third and fourth packages enable WebSphere Partner Gateway to pass E41 messages between partners and back-end.

To upload the CIDX packages:

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Upload/Download Packages**.

3. Select **No** for **WSDL Package**.
4. Click **Browse** and select the RNIF package for communicating with partners.

   The RNIF packages are located, by default, in the B2BIntegrate/rosettanet directory on the installation medium. For example, if you were uploading RNIF Version 2.00 package, you would browse to the B2BIntegrate/rosettanet directory and select: Package_RNIF_V0200.zip.
5. Make sure **Commit to Database** is set to **Yes**.
6. Click **Upload**.
7. Click **Browse** again and select the RNIF package for communicating with back-end applications.

   For example, if you were uploading the RNIF Version 2.00 package, you would browse to the B2BIntegrate/rosettanet directory and select Package_RNSC_1.0_RNIF_V02.00.zip.
8. Click **Upload**.

   The packages needed to communicate with partners or with the back-end system are now installed in the system. If you check the Manage Document Definitions page, you see an entry for **Package: RNIF/Protocol: Rosettanet**, which represents the packaging for communicating with partners, and **Package: Backend Integration/Protocol: RNSC**, which represents the packaging for communicating with back-end applications.
9. For each PIP you want to support, upload the PIP document type package for the PIP and for the RNIF version you are supporting.

   For example, to upload the E41 CIDX PIP (Order Create) to be sent to a partner, perform the following steps:

   a. Click **Browse**, and select **BCG_Package_RNIF1.1_E414.0.zip** from the B2BIntegrate/Rosettanet directory.
   b. Make sure **Commit to Database** is set to **Yes**.
   c. Click **Upload**.

   The E41 PIP now appears as the document type underneath Package: RNIF/Protocol: RosettaNet on the Manage Document Definitions page. An activity, action, and two signals are also displayed. They are included in the upload of the PIP.

   To upload the E41 PIP to be sent to the back-end application, perform the following steps:

   a. Click **Browse**, and select **BCG_Package_RNSC1.0RNIF1.1_E414.0.zip**.
   b. Make sure **Commit to Database** is set to **Yes**.
   c. Click **Upload**.

   The E41 PIP now appears as the document type underneath Package: Backend Integration/Protocol: RNSC on the Manage Document Definitions page.

# Configuring attribute values

## About this task

For RNIF document definitions, most of the values of the attributes are already set and do not need to be configured. However, you do need to set the following attributes:

RNIF (1.1) package

- **GlobalSupplyChainCode** - Identify the type of supply chain used by the partner. The types are Electronic Components, Information Technology, and Semiconductor manufacturing. This attribute does not have a default value.

To set the attributes, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Expand** icons to individually expand a node to the appropriate document definition level or select **All** to expand all displayed document definition nodes.
3. In the **Actions** column, click the **Edit Attribute Values** icon for the package (for example, Package: RNIF (1.1) or Package: RNIF (V02.00)) you want to edit.
4. In the **Document Definition Context Attributes** section, go to the **Update** column of the attribute you want to set and select or type the new value. Repeat for each attribute that you want to set.
5. Click **Save**.

**Note:** You can also update RosettaNet attributes at the connection level by clicking **Attributes** for the source or target and then entering or changing the values in the **Update** column. Refer to "Specifying or changing attributes" on page 232.

# Creating interactions

## About this task

The following process describes how to create an interaction between a back-end system and a partner. Note that you must create an interaction for each PIP that you want to send and one for each PIP that you want to receive.

Before you begin, make sure that the appropriate RNIF document definitions have been uploaded and that the packages for the PIP you want to use have been uploaded.

To create an interaction for a particular PIP, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand the **Source** tree to the **Action** level and expand the **Target** tree to the **Action** level.
4. In the trees, select the document definitions to use for the source context and the target context. For example, if the partner is the initiator of a E41 PIP select the following document definitions:

*Table 17. 3C6 PIP initiated by a partner*

| Source | Target |
|---|---|
| Package:RNIF(1.1) | Package BackEnd Integration(1.1) |
| Protocol:RosettaNet(1.1) | Protocol:RNSC(1.0) |
| Document Type: E41 (4.0) | Document Type: E41 (4.0) |
| Activity: OrderCreate | Activity:OrderCreate |
| Action: Order Create | Action: Order Create |

For a two-action PIP such as 3A4 initiated by a partner, select the following document definitions for the first action:

*Table 18. 3A4 PIP initiated by a partner*

| Source | Target |
|---|---|
| Package: RNIF (V02.00) | Package: Backend Integration (1.0) |
| Protocol: RosettaNet (V02.00) | Protocol: RNSC (1.0) |
| Document Type: 3A4 (V02.02) | Document Type: 3A4 (V02.02) |
| Activity: Request Purchase Order | Activity: Request Purchase Order |
| Action: Purchase Order Request Action | Action: Purchase Order Request Action |

5. In the Action field, select **Bi-Directional Translation of RosettaNet and RosettaNet Service Content with Validation**.
6. Click **Save**.

## Viewing CIDX documents

### About this task

The RosettaNet Viewer displays information about CIDX documents. You can display raw documents and associated document processing details and events using specific search criteria. This information is useful if you are trying to determine whether a document was successfully delivered or to determine the cause of a problem.

To display the RosettaNet Viewer complete the following:

1. Click **Viewers > RosettaNet Viewer**.
2. Select the appropriate search criteria.
3. Click **Search**.

# ebMS documents

The ebMS mechanism provides a standard way to exchange business Messages among ebXML Trading Partners. The ebXML Messaging Service provides a reliable means to exchange business messages without relying on proprietary technologies and solutions. This section shows you how to set up document definitions and interactions for those documents.

## Creating document definitions

### About this task

ebMS messaging requires that a Collaboration Profile Agreement (CPA) XML file be uploaded before documents can be defined.

To upload a CPA XML file, complete the following:

1. Click **Hub Admin** > **Hub Configuration** > **ebMS**.
2. Click **Upload CPA**.
3. Click **Browse** and select the appropriate CPA package.
4. Ensure that **ebMS Version** 2.0 is selected.
5. Click **Upload**.

During the CPA upload process, you will be asked to select the internal partner from the partners present in the CPA. The internal partner is treated as the

manager in the ebMS flow, and all the targets in ebMS flow for the internal partner will use Backend Integration or N/A packaging. However, on console the partner will be shown as external partner only.

The ebMS now appears as a package as well as a protocol under ebMS and Package: Backend Integration on the Manage Document Definitions page.

ebMS flow can also be configured in WebSphere Partner Gateway without CPA. To do so, create ebMS document definitions, B2B capabilities from the WebSphere Partner Gateway console as described in “Overview of document types” on page 99. Actually, while uploading the CPA, all the configurations will be automatically done. In the absence of CPA, follow the steps given in this section.

## Configuring attribute values

### About this task

For ebMS document definitions, most of the values of the attributes are already set and do not need to be configured. However, you do need to set the following attributes:

**ebMS package**

- **Time To Acknowledge in min**- Set the amount of time to wait for an acknowledgment before resending the original request. This attribute works in conjunction with Retry Count. The units are in minutes. The default value is 30.
- **Retry Count** - Set the number of times to send a request if an acknowledgment is not received. This attribute works in conjunction with Time to Acknowledge. The default value is 3.
- **Non-Repudiation Required** - Set whether to store the original document in the non-repudiation store. The default value is Yes.

  **Note:** In WebSphere Partner Gateway 6.2, the non-repudiation information is obtained from the partner connection parameters. The partner connection parameters are obtained after a successful partner connection look-up. By default, non-repudiation is set to "Yes," which means that if the information is not available from the partner connection for some reason, the document will be put in the non-repudiation store.
- **Message Store Required** - Set whether to store the document in the message store. The default value is Yes.

  **Note:** The message store information is obtained from the partner connection parameters. The partner connection parameters are obtained after a successful partner connection look-up. By default, message store is set to "Yes," which means that the document will be persisted in the message store.
- **Non-Repudiation of Receipt** - Set whether to store the receipt in the non-repudiation store. The default value is Yes.
- **Retry Interval** - Set the amount of time the system waits between retry attempts. This attribute works in conjunction with Retry Count. The default is 5 minutes.

To set the attributes, perform the following steps:

1. Click **Hub Admin > Hub configuration > Document Definition**.
2. Click **Expand** icons to individually expand a node to the appropriate document definition level or select **All** to expand all displayed document definition nodes.

3. In the **Actions** column, click the **Edit Attribute Values** icon for the package you want to edit.
4. In the **Document Definition Context Attributes** section, go to the **Update** column of the attribute you want to set and select or type the new value. Repeat for each attribute that you want to set.
5. Click **Save**.

**Note:** You can also update ebMS attributes at the connection level by clicking **Attributes** for the source or target and then entering or changing the values in the **Update** column. Refer to "Specifying or changing attributes" on page 232.

# Creating interactions

## About this task

The following process describes how to create an interaction between a back-end system and a partner.

Before you begin, make sure that the appropriate ebMS document definitions have been uploaded.

To create an interaction for a particular partner, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand the Source tree to the Action level, and expand the Target tree to the Action level.
4. In the trees, select the document definitions to use for the source context and the target context. For example, if the partner is the initiator of an ebMS, select the following document definitions:

*Table 19. ebMS initiated by a partner*

| Source | Target |
|---|---|
| Package: ebMS | Package: Backend Integration (1.0) |
| Protocol: ebMS | Protocol: ebMS |
| Document Type: ALMService | Document Type: ALMService |
| Activity: ALMService | Activity: ALMService |
| Action: Remittance ALMBusiness | Action: ALMBusiness |

If the back-end system is the initiator of the ebMS, select the following document definitions:

*Table 20. ebMS initiated by a back end system*

| Source | Target |
|---|---|
| Package: Backend Integration (1.0) | Package: ebMS |
| Protocol: ebMS | Protocol: ebMS |
| Document Type: ALMService | Document Type: ALMService |
| Activity: ALMService | Activity: ALMService |
| Action: ALMBusiness | Action: Remittance ALMBusiness |

5. Optionally, in the Action field select **ebMS Split and Parse**.

Selecting this handler will extract the payloads from the ebMS message coming from the partner and introduce the payloads back to the flow as if they were coming from the partner separately. This handler should not be selected when the back end system is initiating the message. If you are not selecting this handler, select Pass Through for the action field

6. Click **Save**.

**Note:** In some ebMS flows, for example in STAR specifications, the ebMS Service element (the ebMS Service value is same as the WPG Channel Document Flow Definition value) is not a URI but a string. In such cases, as per ebMS 2.0 spec, a type attribute should be present with the Service element in the ebMS SOAP Message. For example, in a STAR specification, the type attribute should have a value of "STARBOD." You can configure such an attribute on the target side of Document Flow Definition attributes. (See Table 22 on page 136).

# Mapping of ebMS CPA to WebSphere Partner Gateway configuration

## About this task

This section provides mapping between Collaboration Profile Agreement (CPA) and WebSphere Partner Gateway UI configuration. The features are listed along with the corresponding WebSphere Partner Gateway UI configuration.

**1.**

**Feature**
**Element/Attribute**

**1.1 CPAId 1**

**Imported/Manually Configured**: Imported

**WebSphere Partner Gateway UI Configuration:**

CPAID is configured through the channels associated between two partners. You can see the value by navigating to **Hub Admin > ebMS** in WebSphere Partner Gateway console. Click Search and then View details icon from the search results displayed.

**2.**

**Feature**
**Element/Attribute**

**1.2. Status 1**

**Imported/Manually Configured**: Imported but not stored in WebSphere Partner Gateway. Also, this cannot be configured manually.

**WebSphere Partner Gateway UI Configuration:**

This attribute cannot be configured in WebSphere Partner Gateway. The value is checked while importing the CPA. One of the following status is displayed while importing:

- Agreed : The CPA can be imported.

- Signed : The CPA can be imported and the signature is verified before importing.
- Proposed: CPA cannot be imported.

**3.**

**Feature**
**Element/Attribute**

**1.3 Start 1**

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

This attribute cannot be configured in WebSphere Partner Gateway. It can be set only from CPA import. You can see the value by navigating to **Hub Admin > ebMS** in WebSphere Partner Gateway console. Click Search and then View details icon from the search results displayed.

**4.**

**Feature**
**Element/Attribute**

**1.4 End 1**

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

This attribute cannot be configured in WebSphere Partner Gateway. It can be set only from CPA import. You can see the value by navigating to **Hub Admin > ebMS** in WebSphere Partner Gateway console. Click Search and then View details icon from the search results displayed.

**5.**

**Feature**
**Element/Attribute**

**1.5 Conversation Constraints 0, 1 (9.5) - invocationLimit 0,1 - concurrentConversations 0, 1**

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

This attribute cannot be configured in WebSphere Partner Gateway. It can be set only from CPA import. You can see the value by navigating to **Hub Admin > ebMS** in WebSphere Partner Gateway console. Click Search and then View details icon from the search results displayed.

**6.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**
partyName 1

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

To view the values, navigate to **Account Admin > Profiles > Partner**. Click Search and then View details icon from the search results displayed for the partner in CPA.

**7.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**
defaultMshChannelId 1

**Imported/Manually Configured**: Imported but not stored in WebSphere Partner Gateway. Also, this cannot be configured manually.

**WebSphere Partner Gateway UI Configuration:**

The values are used while importing the CPA to set channel attributes for **Activity-MSHService** signal elements like Ping, Status request, MessageError, and Acknowledgment. These channel values are again over ridden if there exists any "OverrideMshActionBinding" element in CPA for any specific action element.

**8.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**
defautMshPackageId 1

**Imported/Manually Configured**: Imported but not stored in WebSphere Partner Gateway. Also, this cannot be configured manually.

**WebSphere Partner Gateway UI Configuration:**

The values are used while importing the CPA to set channel attributes for **Activity-MSHService** signal elements like Ping, Status request, MessageError, and Acknowledgment. These channel values are again over ridden if there exists any "OverrideMshActionBinding" element in CPA for any specific action element.

**9.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**
PartyId 1, *

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

To view the values, navigate to **Account Admin > Profiles > Partner**. Click Search and then View details icon from the search results displayed for the partner in CPA.

**10.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**
type

**Imported/Manually Configured**: Not imported and cannot be configured.

**11.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**

```
PartyRef 1,*= (8.4.2)
- xlink:type F
- xlink:href 1
- type Fixed
- schemaLocation Implied
```

**Imported/Manually Configured**: Not imported and cannot be configured.

**12.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**
1.6.3 CollaborationRole 1,*

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

WebSphere Partner Gateway supports multiple collaboration role elements.

**13.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**

```
.6.3.1 ProcessSpecification 1
- name 1
- version 1
- xlink:type 1
- xlink:href
1 - uuid ImpliedReference 0,*  (8.4.4.6)
- URI 0, 1
 Transforms 1
  Transform
1 - Algorithm Fixed
 DigestMethod 1
  DigestValue 1
```

**Imported/Manually Configured**: Not imported.

**WebSphere Partner Gateway UI Configuration:**

Cannot be configured.

**14.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**

```
1.6.3.2 Role 1   (8.4.5)
- name 1
- xlink:type Fixed
- xlink:href 1
```

**Imported/Manually Configured**: The attribute **xlink:href** is imported, other attributes are not imported.

**WebSphere Partner Gateway UI Configuration:**

The value can be configured in channel attributes **Account Admin > Connections > Partner Connections**. Search for the channels and access the channel attribute - **Role**.

**15.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**

```
1.6.3.3  ApplicationCertificateRef 0,1  (8.4.6)
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

The value cannot be configured. The certificate specified for the attribute **certId** is loaded into the file system but not into WebSphere Partner Gateway.

**16.**

**Feature**
**Element/Attribute**

**1.6 PartyInfo 2**

```
1.6.3.4  ApplicationSecurityDetailsRef 0, 1 (8.4.7)
- securityId 1
```

**Imported/Manually Configured**: Not imported.

**WebSphere Partner Gateway UI Configuration:**

Cannot be configured.

**17.**

**Feature**
**Element/Attribute**

**1.6.3.5 ServiceBinding 1**

```
1.6.3.5.1 Service 1 (8.4.9)
- type  Implied
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

- **Service** : is the name of the document definition. To view the value, navigate to **Hub Admin > Document Definitions**. The Service value will be displayed as Document Type and Activity under ebMS package and Backend integration Package.
- **Type**: Type is used as channel attribute under **Account Admin > Connections > Partner Connections**. Search for the channels and access the channel attribute **Service Type**.

**18.**

**Feature**

**Element/Attribute**

**1.6.3.5 ServiceBinding 1**

```
1.6.3.5.1 Service 1 (8.4.9)
- type  Implied
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

- **Service** : is the name of the document definition. To view the value, navigate to **Hub Admin > Document Definitions**. The Service value will be displayed as Document Type and Activity under ebMS package and Backend integration package.
- **Type**: Type is used as channel attribute under **Account Admin > Connections > Partner Connections**. Search for the channels and access the channel attribute **Service Type**.

**19.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
ThisPartyActionBinding 1
- action 1
- packageId 1
- xlink:href  Implied -
xlink:type Fixed
BusinessTransactionCharacteristics 1
- isNonRepudiationRequired
  All implied
isNonRepudiationReceiptRequired
- isConfidential
- isAuthenticated
- isAuthorizationRequired
- isTamperProof
- isIntelligibleCheckRequired
- timeToAcknowledgeReceipt
- timeToAcknowledgeAcceptance
- timeToPerform
- retryCountChannelId 1,*
ActionContext 0, 1
- binaryCollaboration 1
- businessTransactionActivity 1
- requestOrResponseAction 1
CollaborationActivity 0, 1
- name 1
OtherPartyActionBinding 0, 1
CanReceive 0, 1
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

- **CanSend** – A channel is created from **Backend integration > ebMS > Service name > Action of the partnerA** to **ebMS > Service name > Action of partnerB** (partnerB having the **CanReceive** element that is bound through **OtherPartyActionBinding** element).
- **Action** – Imported and created as an Action element under **Activity** in document definition.
- **packageId** – The referencing package ID attributes are stored as channel attributes.
- Xlink:href and xlink:type: Not imported and cannot be configured.
- **isNonRepudiationRequired, isNonRepudiationReceiptRequired, isIntelligibleCheckRequired, timeToAcknowledgeReceipt, timeToPerform**: These attributes are configured as channel attributes.
- **isConfidential, isAuthenticated, isTamperProof, isAuthorizationRequired, timeToAcknowledgeAcceptance, retryCount** - Are not imported and not configurable.
- **ChannelId** 1, * : Only one value is accepted for WebSphere Partner Gateway. The referencing attributes are set as channel attributes.
- **binaryCollaboration, businessTransactionActivity, requestOrResponseAction, CollaborationActivity** – Are not imported and not configurable.
- **OtherPartyActionBinding** - Imported. The reference is used to create the channel.
- **CanReceive** - Imported and is treated as synchronous if there exists any other channel for the same connection.

**20.**

**Feature**
**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

1.6.3.5.3 CanReceive 0, * (8.4.11)
ThisPartyActionBinding 1
OtherPartyActionBinding 0, 1
CanSend 0, 1

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

- **CanReceive** – A channel is created from **ebMS > Service name > Action of the partnerA** to **Backend Integration > ebMS > Service name > Action of partnerB** (partnerB having the **CanSend** element bound through **OtherPartyActionBinding** element).
- **OtherPartyActionBinding** - Imported. The reference is used to create the channel.
- **CanSend** - Imported and is treated as synchronous if there exists any other channel for the same connection.

**21.**

**Feature**
**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.4 Certificate 1, * (8.4.18)
- certId  KeyInfo
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

The certificates are stored in the file system and it must be manually loaded in WebSphere Partner Gateway under **Account Admin > Profiles > Certificates**.

**22.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.5 SecurityDetails 0, *   (8.4.18)
- securityId 1 TrustedAnchor 0, *
AnchorCertificateRef 1, *
SecurityPolicy 0, 1
```

**Imported/Manually Configured**: Not imported. Only the reference certificates are loaded into the file system.

**23.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.6 DeliveryChannel 1, * (8.4.22)
- channelId 1
- transportId 1
- docExchangeId1
MessagingCharacteristics 1
- syncReplyMode All implied
- ackRequested attribute
- ackSignatureRequested
- duplicateElimination
- actor
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

- **channelId** : The referencing attributes are set as channel attributes.
- **transportId**: The referencing attributes are used to create the gateway and set as default gateway for the channel.
- **docExchangeId**: The referencing attributes are set as channel attributes.
- **syncReplyMode, ackRequested, ackSignatureRequested, duplicateElimination, actor** : These attributes are imported and configured as channel attributes.

**24.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.7 Transport 1, * (8.4.24)
- transportId 1
TransportSender 0, 1 (8.4.25)
TransportProtocol 1
- version 1
ImpliedAccessAuthentication 0, *
TransportClientSecurity 0, 1
TransportSecurityProtocol 1
- version 1
ImpliedClientCertificateRef  0, 1
- certId 1
ServerSecurityDetailsRef 0, 1
- securityId 1
EncryptionAlgorithm 0, *
- minimumStrength All Implied
- oid
- w3c
- enumeratedType
```

**Imported/Manually Configured**: Not imported.

**25.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.7 Transport 1, * (8.4.24)
TransportReceiver 0, 1 (8.4.33)
TransportProtocol 1
- version 1
ImpliedEndpoint 1, *
- uri 1
- type  ImpliedAccessAuthentication 0, *
TransportServerSecurity 0, 1
TransportSecurityProtocol 1
- version 1
ServerCertificateRef 1
- certId 1
ClientSecurityDetailsRef 0, 1
- SecurityId 1
EncryptionAlgorithm 0, *
- minimumStrength All Implied
- oid
- w3c
- enumeratedType
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

- **Transport Protocol** : Defines the gateway protocol.
- **Version** : Defines the gateway protocol version.
- **URL**: Defines the gateway URL. These values can be seen under **Account Admin > Profiles > PartnerSearch**. For all partners and for the selected partner, click **Destinations** tab. Remaining attribute values are not imported.

**26.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.8 DocExchange (8.4.39)
- docExchangeId 1 1.6.8.2.1
ebXMLSenderBinding 0, 1 (8.4.40)
- version ReliableMessaging 0, 1
Retries 0, 1
RetryInterval 0, 1
MessageOrderSemantics 1
PersistDuration 0, 1
SenderNonRepudiation 0, 1
NonRepudiationProtocol 1
- version 1 Implied
HashFunction 1
SignatureAlgorithm 1
- oid All implied
- w3c
- enumeratedType
SigningCertificateRef 1
- certId 1
SenderDigitalEnvelope 0, 1
DigitalEnvelopeProtocol 1
- version 1  EncryptionAlgorithm 1
- minimumStrength All Implied
- oid
- w3c
- enumeratedType
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

**Retries, RetryInterval, MessageOrderSemantics, PersistDuration, HashFunction, SignatureAlgorithm, DigitalEnvelopeProtocol, EncryptionAlgorithm** : These values are imported and stored as channel attributes, present in **Account Admin > Connections > Partner Connections**. Search for the channels and go to **Channel Attributes**. Remaining values are not imported and cannot be configured.

**27.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.8.2 ebXMLReceiverBinding 0, 1 (8.4.53)
- version 1
ReliableMessaging 0, 1
Retries 0, 1
RetryInterval 0, 1
MessageOrderSemantics 1
ReceiverNonRepudiation 0, 1
NonRepudiationProtocol 1
- version 1
HashFunction 1
SigningAlgorithm 1
- oid All Implied
- w3c
- enumeratedType
SigningSecurityDetailsRef 1
- securityId 1
ReceiverDigitalEnvelope 0, 1
DigitalEnvelopeProtocol 1
- version 1
EncryptionAlgorithm 1
- minimumStrength All Implied
- oid
- w3c
```

```
- enumeratedType
EncryptionCertificateRef 1
- certId 1
NamespaceSupported 0, *
- location 1
- version Implied
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

**Retries, RetryInterval, MessageOrderSemantics, PersistDuration, HashFunction, SignatureAlgorithm, DigitalEnvelopeProtocol, EncryptionAlgorithm** : These values are imported and stored as channel attributes, present in **Account Admin > Connections > Partner Connections**. Search for the channels and go to **Channel Attributes**. Remaining values are not imported and cannot be configured.

**28.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.6.9  OverrideMshActionBinding 0, *  (8.4.58)
- action 1
- channelId
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

For the specified action, the channel attributes are set using the reference channel ID.

**29.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.7 SimplePart (8.5)
- id 1
- mimetype 1
- mimparameters  Implied
- xlink:role
 ImpliedNamespaceSupported 0, *
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

**Mimetype** : Values are imported and stored as channel attributes. Remaining values are not imported and cannot be configured.

**30.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.8 Packaging (8.6)
- id 1
ProcessingCapabilities 1, *
- parse 1
- generate 1
CompositeList 0, *
Composite 0, *
- mimetype 1
- id 1
- mimeparameters ImpliedConstituent 1, *
- idref 1
- excludeFromSignature Implied
- minOccurs Implied
- maxOccurs Implied
SignatureTransform 0, 1
Transform 1, *
EncryptionTransform 0, 1
Transform 1, *
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

**Composite : mimetype, mimeparameters, Constituent-idref, Constituent-excludeFromSignature, signatureTransform, encryptionTransform, Algorithm**: These values are imported and stored as channel attributes in **Account Admin > Connections > Partner Connections**. Search for the channels and go to **Channel Attributes**. Remaining values are not imported and cannot be configured.

**31.**

**Feature**
**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
Encapsulation 0, *
- mimetype 1
- id 1
- mimeparameters  ImpliedConstituent 1
- idref 1
- excludeFromSignature Implied
- minOccurs Implied
- maxOccurs Implied
SignatureTransform 0, 1
Transform 1, *
EncryptionTransform 0, 1
Transform 1, *
```

**Imported/Manually Configured**: Imported.

**WebSphere Partner Gateway UI Configuration:**

**Encapsulation : mimetype, mimeparameters, Constituent-idref, Constituent-excludeFromSignature, signatureTransform, encryptionTransform, Algorithm**: These values are imported and stored as channel attributes in **Account Admin > Connections > Partner Connections**. Search for the channels and go to **Channel Attributes**. Remaining values are not imported and cannot be configured.

**32.**

**Feature**
**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.9 Signature 0, 1 (8.7)
ds:Signature 1,3
SignedInfo 1
CanonicalizationMethod 0, 1
SignatureMethod 1
- AlgorithmReference 1, *
- URI FixedTransforms 1
Transform 1
- Algorithm Fixed
```

**Imported/Manually Configured**: Not Imported.

**WebSphere Partner Gateway UI Configuration:**

Cannot be configured.

**33.**

**Feature**

**Element/Attribute**

**1.6.3.5.2 CanSend 0, * (8.4.10)**

```
1.10 Comments 0, * (8.8)
- xml:lang
```

**Imported/Manually Configured**: Not Imported.

**WebSphere Partner Gateway UI Configuration:**

Cannot be configured.

## Connection Attributes

The following table provides the routing object attributes, which can be seen in the business channels of the message on the ebMS packaging.

Click **Account Admin > Connections > Partner Connections** and select Source and Target. If the channel is for inbound ebMS message, click **Attributes** of source side, else if the channel is for outbound ebMS message, click **Attributes** of the target side. Scroll down the resulting screen and click **Action** folder.

*Table 21. Connection Attributes*

| CPA XML Attributes | Default value | Possible values | Display text in WebSphere Partner Gateway |
|---|---|---|---|
| isNonRepudiationRequired | False | True/false - mapped as Yes/No | Non-Repudiation Required |
| isNonRepudiationReceiptRequired | False | True/false - mapped as Yes/No | Non-Repudiation of Receipt |
| timeToAcknowledgeReceipt | | | Time To Acknowledge |
| Retries | 3 | Some Number | Retry Count |
| MessageOrderSemantics | Not Guaranteed | "Guaranteed" "NotGuaranteed" | Message Order Semantics |
| PersistDuration | P1D | | Persist Duration |

*Table 21. Connection Attributes (continued)*

| CPA XML Attributes | Default value | Possible values | Display text in WebSphere Partner Gateway |
|---|---|---|---|
| syncReplyMode | None | "mshSignalsOnly"<br><br>"signalsOnly"<br><br>"responseOnly"<br><br>"signalsAndResponse"<br><br>"none"<br><br>(Moved to phase 2) | Sync Reply Mode |
| ackRequested | Per Message | "always" - implies that acknowledgment should always be requested.<br><br>"never" - implies that acknowledgment should never be requested.<br><br>“perMessage” - implies that acknowledgment can or cannot be requested depending on the ack element present in ebXML document. | Acknowledgment Requested |
| ackSignatureRequested | Per Message | "always"<br><br>"never"<br><br>"perMessage" | Acknowledgment Signature Requested |
| duplicateElimination | Per Message | "always"<br><br>"never"<br><br>"perMessage" | Duplicate Elimination |
| Actor | "urn:oasis:names:tc:ebxml-msg:actor:nextMSH" | "urn:oasis:names:tc:ebxml-msg:actor:nextMSH""urn:oasis:names:tc:ebxml-msg:actor:toPartyMSH" | Actor |
| PartyRole | - | Role in CPA | Role |
| Retry Interval | 270 | - | Retry Interval |
| NonRepudiationProtocol | - | http://www.w3.org/2000/09/xmldsig# | Signing Protocol |
| SignatureAlgorithm | - | 1. http://www.w3.org.2000/09/xmldsig#dsa-sha1<br><br>2. http://www.w3.org/2000/09/xmldsig#rsa-sha1<br>**Note:** In ebMS, hmac-sha1 is not supported. | Signature Algorithm |
| isEncryptionRequired | No | True/false - mapped as Yes/no | EncryptionRequired |
| isCompressionRequired | No | True/false - mapped as Yes/no | Compression Required |
| /Packaging/CompositeList /Encapsulation/Constituent: mimetype | - | | Compress Mimeype |
| /tp:SenderDigitalEnvelope /tp:DigitalEnvelopeProtocol | - | SMIME XMLEncryption | EncryptionProtocol |
| /tp:SenderDigitalEnvelope /EncryptionAlgorithm | - | 3des-cbc, aes128-cbc, aes-256-cbc | Encryption Algorithm |

*Table 21. Connection Attributes (continued)*

| CPA XML Attributes | Default value | Possible values | Display text in WebSphere Partner Gateway |
|---|---|---|---|
| /tp:ReceiverDigitalEnvelope /tp:DigitalEnvelopeProtocol | - | SMIME XMLEncryption | EncryptionProtocol |
| /tp:ReceiverDigitalEnvelope /EncryptionAlgorithm | - | 3des-cbc, aes128-cbc, aes-256-cbc | Encryption Algorithm |
| /Packaging/CompositeList /Encapsulation tp:MimeType | - | text/xml application/pkcs7-mime | Encryption Mime Type |
| /Packaging/CompositeList /Encapsulation- tp:mimeparameters | - | | Encryption Mime Parameter |
| /Packaging/CompositeList /Encapsulation/Constituent: mimetype | - | | Encryption Constituent |
| /Packaging/CompositeList / Composite/ tp:mimeparameters | - | | Package Mime Parameter |
| /Packaging/CompositeList / Composite /Constituent: mimetype | - | | PackagingConstituent |
| /Packaging/CompositeList /Composite/Contituent /excludeFromSignature: mimetype | - | | Exclude from Signature |
| /Packaging/CompositeList / Composite/Contituent/ SignatureTransform | - | 1. BASE64<br>2. ENVELOPED<br>3. XPATH<br>4. XPATH2<br>5. XSLT | Signature Transformation Algorithm |
| /Packaging/CompositeList /Composite/Contituent/ EncryptionTransform | - | 1. BASE64<br>2. ENVELOPED<br>3. XPATH<br>4. XPATH2<br>5. XSLT | Encryption Transformation Algorithm |

### Limitations

The following are the limitations of mapping CPA to WebSphere Partner Gateway:

1. Certificates from CPA are not imported into the WebSphere Partner Gateway. They are stored in the file system and the administrator has to manually verify those certificates and upload them to WebSphere Partner Gateway.
2. WebSphere Partner Gateway can address the synchronous and asynchronous flows from CPA, but not multiple bindings having the same action value.
3. Only 9 digit numeric DUNS ID is supported (freeform is not supported).

## Mapping of ebMS SOAP headers to WebSphere Partner Gateway headers

The ebMS spec 2.0 defines a set of mandatory headers required to be present in ebMS SOAP message. The below table will give the mapping between some of these ebMS mandatory headers and WebSphere Partner Gateway headers from which their values are taken.

*Table 22. ebMS SOAP headers and corresponding WebSphere Partner Gateway headers*

| Serial No. | Header name in ebMS SOAP Message | Corresponding Header name in WebSphere Partner Gateway |
|---|---|---|
| 1 | From PartyId | "x-aux-sender-id" set by backend system |
| 2 | From Role | Role attribute on the Source side of Document Definition attributes |
| 3 | From PartyId Type | User cannot configure it. If the PartyId is DUNS, the "type" value will be "urn:duns." Otherwise, it will be "string." |
| 4 | To PartyId | "x-aux-receiver-id" set by backend system |
| 5 | To Role | Role attribute on the Target side of Document Definition attributes. |
| 6 | To PartyId Type | User cannot configure it. If the PartyId is duns then "type" value will be "urn:duns" else it will be "string" |
| 7 | CPAId | If a CPA is present in database, then WebSphere Partner Gateway will use the CPA-ID present in the CPA. Otherwise, the user can configure the CPA ID attribute present on the Target side of Document Definition attributes. If the user has not configured this attribute and a CPA is not present, then WebSphere Partner Gateway will generate a CPA ID based on the partner IDs. |
| 8 | Conversation Id | "x-aux-process-instance-id" set by backend system. If backend system does not set it, then WebSphere Partner Gateway will generate its own Conversation ID. |
| 9 | Service | The Document Definition value on the Target partner connection.<br>**Note:** The Document Definition and Activity will be same in an ebMS flow. |
| 10 | Service Type | ServiceType attribute on the Target side of Document Definition attributes |
| 11 | Action | The Action value on the Target partner connection |
| 12 | MessageId | "x-aux-msg-id" set by backend system. If backend system does not set it, then WebSphere Partner Gateway will generate its own Message ID. |

If you are sending an ebMS synchronous response to an ebMS request document, the backend system needs to set "x-aux-request-msg-id" header on the response document. The value of this header will be the message ID of the request message. Moreover, the response document should be in the same conversation as the request document. This means that the "x-aux-process-instance-id" for the response should be same as ConversationId of the request.

The ConversationId and MessageId of the request document are sent to backend as "x-aux-process-instance-id" and "x-aux-msg-id" respectively.

# Viewing ebMS documents

## About this task

The ebMS Viewer displays information about ebMS documents. You can display raw documents and associated document processing details and events using

specific search criteria. This information is useful if you are trying to determine whether a document was successfully delivered or to determine the cause of a problem.

To display the ebMS Viewer complete the following:

1. Click **Viewers > ebMS Viewer**.
2. Select the appropriate search criteria.
3. Click **Search**.

In the ebMS Viewer, documents are organized based on conversation ID. This means that all the documents with the same Conversation ID will be grouped together and can be seen by clicking the More details icon on the left side of each row of conversation ID. When you click the More details icon, a new page displays showing all the messages in that conversation. On the top of the page, there is a attribute called "Conversation Status." The value of this attribute is the next message expected in that conversation.

## Requesting status for an ebMS message

### About this task

To request the status of an ebMS message, complete the following steps:

1. After you have found the ebMS document you are interested in, click the **View details** icon next to it.
2. Click Request Status. The status of that document then displays.

To refresh the status, click **View Status**.

When you configure for ebMS Status Request and Status Response documents, consider the following:

- Only the Status Request connection needs to be created. The Status Response connection will use the existing Status Request connection.
- For a Status Request connection from the internal partner to an external partner, the connection's Source Destination is not used.
- For a Status Request connection from an external partner to an internal partner, the connection's Source Destination is used for sending the reply Status Response document back to the external partner.
- If a user is not having a CPA, then he or she needs to enable B2B Capabilities and create channel for ebMS Status Request message as follows:
  - For inbound ebMS Status Request Message,

    The source side B2B capability should be:

    Package: N/A (N/A)
    Protocol: ebMS (2.0)
    Document Type: MSHService (2.0)
    Activity: MSHService (2.0)
    Action: StatusRequest(N/A)

    The target side B2B capability should be

    Package: ebMS (2.0)
    Protocol: ebMS (2.0)
    Document Type: MSHService (2.0)
    Activity: MSHService (2.0)
    Action: StatusRequest(N/A)

– For outbound ebMS Status Request Message

  The source side B2B capability should be:

  Package: ebMS (2.0)
  Protocol: ebMS (2.0)
  Document Type: MSHService (2.0)
  Activity: MSHService (2.0)
  Action: StatusRequest(N/A)

  The target side B2B capability should be

  Package: N/A (N/A)
  Protocol: ebMS (2.0)
  Document Type: MSHService (2.0)
  Activity: MSHService (2.0)
  Action: StatusRequest(N/A)

The user should then activate the channel and set destinations from the partner connection page.

**Note:** This information holds true for ebMS error and Acknowledgment. The action for these channels will change to MessageError and Acknowledgment respectively.

# Pinging ebMS partners

## About this task

From the Test Partner connection page, you can ping ebMS partners. This means that you can send a ping message to a partner, and, if the partner is up and ready to receive, the partner responds with a pong message. Once you upload a CPA, the ping-pong channel is created.

For the Ping to work connections have to be defined with the partner involved. For details, see the section for pinging ebMS partners in the *WebSphere Partner Gateway Hub Configuration Guide*.

To ping an ebMS partner, complete the following steps:

1. Click **Tools** > **Test Partner Connection**.
2. For **Command**, select **PING ebMS**.
3. Select **From Partner** and **To Partner**.
4. Optionally, select a **Destination** or type a **URL**.
5. Click **Test** to send a ping message.

To determine the status of the ping message, click **Ping Status**. The status for the last ping request then displays under Results.

**Note:** The last ping request may have been initiated from the Test Partner Connection or from a Document Viewer re-send of an existing Ping document.

# Web services

A partner can invoke a Web service hosted by the internal partner. Similarly, the internal partner can invoke a Web service hosted by a partner. The partner or internal partner invokes the Web service via WebSphere Partner Gateway server. WebSphere Partner Gateway acts as a proxy, passing the Web service request to the Web service provider and returning the response synchronously from the provider to the requester.

This section contains the following information for setting up a Web service for use by a partner or an internal partner:

- Identifying the partners for a Web service.
- Setting up a document definition for a Web service.
- Adding document definitions to partner B2B capabilities.
- Restrictions and limitations of Web service support.

## Identifying the partners for a Web service

When a Web service is provided by the internal partner for use by partners, WebSphere Partner Gateway requires identification of both internal and external partners. In WebSphere Partner Gateway you can create multiple internal partners out of which one is set as a default internal partner. To override the default internal partner and select an internal partner, send additional parameters to the WebSphere Partner Gateway receiver, such as **FromPartnerBusinessId** or **ToPartnerBusinessId** depending on outbound or inbound flow respectively. The error condition is that if two different external partner IDs are provided through Basic authentication and URL, then Basic authentication takes precedence. The various possible query strings for outbound flow are: `<Receiver-URL>?to=<business id>` and `<Receiver-URL?to=<business id>&from=<business id>`. The various possible query strings for inbound flow are:`<Receiver-URL` and `Receiver-URL?to=business id`. In case of inbound the **basic authentication** is mandatory.

## Creating document definitions

To set up the document definition, you either upload the WSDL (Web Service Definition Language) files that define the Web service, or you enter the equivalent document definitions manually through the Community Console.

### Uploading the WSDL files for a Web service

#### About this task

The definition for a Web service should be contained in a primary WSDL file, with extension `.wsdl`, which might import additional WSDL files through the `import` element. If there are imported files, these can be uploaded with the primary file using one of the following methods:

- If the file path or (HTTP) URL in the `location` attribute of each `import` element is reachable from the Community Console's server (not the user's machine), the primary file can be uploaded directly and the imported files will be uploaded automatically.
- If all the imported files and primary file are zipped into one file, each with a path corresponding to the path (if any) in the import `location` attribute, uploading the zipped file will upload all the contained primary and imported WSDL files.

For example, suppose the primary WSDL file `helloworldRPC.wsdl` contains the following import element:

```
<import namespace="http://www.helloworld.com/wsdl/helloRPC.wsdl" location=
"bindingRPC.wsdl"/>
<import namespace="http://www.helloworld.com/wsdl/helloRPC.wsdl" location=
"port/porttypeRPC.wsdl"/>
```

And suppose the imported WSDL file `bindingRPC.wsdl` contains the following import element:

The file should contain the following:

```
Name                    Path

helloworldRPC.wsdl

bindingRPC.wsdl

porttypeRPC.wsdl                port\
```

When a WSDL file definition of a Web service is uploaded, the original WSDL is saved as a validation map. (Web service messages are not actually validated against WSDL by WebSphere Partner Gateway.) This is called the *private* WSDL.

In addition, a public WSDL is saved with the private URL replaced by the target URL specified on the Upload/Download Packages page. The public WSDL will be provided to the users of the Web service, who will invoke the Web service at the target's URL (the public URL). WebSphere Partner Gateway will then route the Web service request to a destination that is the original Web service provider's private URL. WebSphere Partner Gateway acts as a proxy, forwarding the Web service request to a private provider URL, which is hidden from the Web service user.

Both the private and public WSDLs (including any imported files) can be downloaded from the Community Console after the WSDL has been uploaded.

**Uploading WSDL files using the Community Console:** WebSphere Partner Gateway provides a way to import WSDL files. If a Web service is defined in a single WSDL file, you can upload the WSDL file directly. If the Web service is defined using multiple WSDL files (this happens when you have imported WSDL files within a primary WSDL file), they would be uploaded in a zipped archive.

**Important:** The WSDL files within the zipped archive must be within a directory specified in the WSDL import element. For example, suppose you have the following import element:

```
<import namespace="http://www.helloworld.com/wsdl/helloRPC.wsdl"
location="path1/bindingRPC.wsdl"/>
```

The directory structure within the zipped archive would be: `path1/bindingRPC.wsdl`.

Now consider this example:

```
<import namespace="http://www.helloworld.com/wsdl/helloRPC.wsdl"
 location="bindingRPC.wsdl"/>.
```

The `bindingRPC.wsdl` file would be at the root level within the zipped archive.

To upload a single WSDL file or zipped archive, use the following procedure.

1. Click **Hub Admin** > **Hub Configuration** > **Document Definition**.
2. Click **Upload/Download Packages**.
3. For **WSDL Package**, click **Yes**.
4. For **Web Service Public URL**, perform one of the following steps:
   - For a Web service provided by the internal partner (which will be invoked by a partner), type the public URL of the Web service. For example:

     `https://<target_host:port>/bcgreceiver/Receiver`

     The URL is typically the same as the production HTTP target defined in Targets.
   - For a Web service provided by a partner (which will be invoked by the internal partner), type the public URL of the partner with a query string. For example:

     `https://<target_host:port>/bcgreceiver/Receiver?to=<partner_business_ID>`
5. Click **Browse** and select the WSDL file or zipped archive.
6. For **Commit to Database**, select **No** if you want to upload the file in test mode. When you select **No**, the file will not be installed into the system. Use the system-generated messages displayed in the Messages box to troubleshoot upload errors. Select **Yes** to upload the file into the system database.
7. For **Overwrite Data**, select **Yes** to replace a file currently in the database. Select **No** to add the file to the database.
8. Click **Upload**. The WSDL file is installed into the system.

**Validating packages using schema files:** A set of XML schemas that describe the XML files that can be uploaded through the console is provided on the WebSphere Partner Gateway installation medium. Uploaded files are validated against these schemas. The schema files are a useful reference for determining the cause of an error when a file cannot be uploaded because of non-conforming XML. The files are: `wsdl.xsd`, `wsdlhttp.xsd`, and `wsdlsoap.xsd`, which contain the schema describing valid Web Service Definition Language (WSDL) files.

The files are located in: `B2BIntegrate\packagingSchemas`

## Creating the document definition manually

To enter the equivalent document definitions manually, follow the procedures in this section. You must also create the Document Type, Activity, and Action entries individually under **Protocol: Web Service**, paying particular attention to the requirements for the Action and its relationship to the received SOAP messages.

In terms of the Package/Protocol/Document Type/Activity/Action hierarchy of document definitions, a supported Web service is represented as:

- **Package: None**
- **Protocol: Web Service (1.0)**
- **Document Type:** {*<Web_service_namespace>*:*<Web_service_name>*} (name and code), which is required to be unique among document types for the Web Service protocol. This is typically the WSDL's namespace and name.
- **Activity:** One activity for each Web service operation, with name and code:

  {*<operation_namespace>*}:*<operation_name>*
- **Action**: One action for the input message of each operation, with name and code:

  {*<namespace_of_identifying_xml_element = namespace_of_first_child_of_soap:body>*}:
  *<name_of _identifying_xml_element = name_of_first_child_of_soap:body>*

The critical definitions are the Actions because WebSphere Partner Gateway will use an Action's namespace and name to recognize an incoming Web service request SOAP message and route it appropriately based on a defined partner connection. The namespace and name of the first child XML element of the received SOAP message's `soap:body` element must match a known Action's namespace and name in WebSphere Partner Gateway's document definitions.

For example, suppose a Web service request SOAP message for a document-literal SOAP binding is:

```
<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance">
 <soapenv:Body>
  <nameAndAddressElt xmlns="http://www.helloworld.com/xsd/helloDocLitSchema">
   <titleElt xmlns="">Mr</titleElt>
   <nameElt xmlns="">Joe Smith</nameElt>
   <addressElt xmlns="">
    <numberElt>123</numberElt>
    <streetElt>Elm St</streetElt>
    <cityElt>Peoria</cityElt>
   </addressElt>
  </nameAndAddressElt>
 </soapenv:Body>
</soapenv:Envelope>
```

WebSphere Partner Gateway would look for a defined Web Service Action with this code:

```
{http://www.helloworld.com/xsd/helloDocLitSchema}:nameAndAddressElt
```

For an RPC binding style SOAP request message, for example:

```
<?xml version="1.0" encoding="UTF-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd=
"http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance">
 <soapenv:Body>
  <ns1:helloWorldRPC soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding/ xmlns:ns1="http://www.helloworld.com/helloRPC">
   <name xsi:type="xsd:string">Joe Smith</name>
  </ns1:helloWorldRPC>
 </soapenv:Body>
</soapenv:Envelope>
```

WebSphere Partner Gateway would look for a defined Web Service action with this code:

```
{http://www.helloworld.com/helloRPC}:helloWorldRPC
```

For an RPC binding, the namespace and name of the first child element of the `soap:body` of a SOAP request message should be the namespace and name of the applicable Web service operation.

For Document-Literal binding, the namespace and name of the first child element of the `soap:body` of a SOAP request message should be the namespace and name of the XML `element` attribute in the `part` element of the input `message` definition for the Web service.

## Creating interactions

### About this task

To create an interaction for a Web service, you use the same Web service document type action for both the Source and the Target.

To create interactions, use the following procedure.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Under **Source**, expand **Package: None > Protocol: Web Service > Document Type: <** *document type*> **> Action:** <*action*>.
4. Repeat the previous step in the **Target** column.
5. Select **Pass Through** from the **Action** list at the bottom of the page. (**Pass Through** is the only valid option supported in WebSphere Partner Gateway for a Web service.)

## Restrictions and limitations of Web service support

WebSphere Partner Gateway supports the following standards:

- WSDL 1.1
- SOAP 1.1
- WS-I Basic Profile v. 1.0 (which contains important restrictions on the form of SOAP messages for document-literal binding)

**Note:**

- WebSphere Partner Gateway has partial support for Basic Profile 1.0.
- SOAP/HTTP binding are supported.
- Rebinding is not supported.
- RPC-encoded/RPC-literal and document-literal binding styles are supported (subject to the restrictions in the WS-I Basic Profile).

See "SOAP Envelope Validate" on page 95 and "SOAP De-envelope" on page 96.

# cXML documents

The WebSphere Partner Gateway Document Manager identifies a cXML document by the root element name of the XML document, which is `cXML` and the version identified by the cXML DOCTYPE (DTD). For example, the following DOCTYPE is for cXML Version 1.2.009:

```
<!DOCTYPE cXML SYSTEM "http://xml.cXML.org/schemas/cXML/1.2.009/cXML.dtd">
```

The Document Manager performs the DTD validation on cXML documents; however, WebSphere Partner Gateway does not provide cXML DTDs. You can download them from www.cxml.org and then upload them into WebSphere Partner Gateway through the Validation Map module in the Community Console. After you upload the DTD, associate it with the cXML document type. Refer to "Associating maps with document definitions" on page 158 for more information about associating the DTD with the cXML document type.

The Document Manager uses two attributes of the cXML root element for document management: the payloadID and timestamp. The cXML payloadID and

timestamp are used as the document ID number and document timestamp. Both are viewable in the Community Console for document management.

The From and To elements within the cXML header contain the Credential element that is used for document routing and authentication. The following example shows the From and To elements as the source and destination of the cXML document.

**Note:** Here and throughout this book, all DUNS numbers are meant to be examples only.

```
<Header>
<From>

            <Credential domain="AcmeUserId">
                <Identity>admin@acme.com</Identity>
            </Credential>
            <Credential domain="DUNS">
                <Identity>130313038</Identity>
            </Credential>
</From
<To>
            <Credential domain="DUNS">
                <Identity>987654321</Identity>
            </Credential>
            <Credential domain="IBMUserId">
                <Identity>test@ibm.com</Identity>
            </Credential>
</To>
```

If more than one credential element is used, the Document Manager uses the DUNS number as the Business Identifier for routing and authentication. In the case where there is no DUNS number given, the first Credential is used.

WebSphere Partner Gateway does not use the information in the Sender element.

In a synchronous transaction, the From and To header is not used in a cXML response document. The response document is sent through the same HTTP connection that is established by the request document.

## cXML document types

A cXML document can be one of three types: Request, Response, or Message.

### Request

There are many types of cXML requests. The `Request` element within the cXML document corresponds to the Document Type in WebSphere Partner Gateway. Typical request elements are:

- OrderRequest
- ProfileRequest
- PunchOutSetupRequest
- StatusUpdateRequest
- GetPendingRequest
- ConfirmationRequest
- ShipNoticeRequest

The following table shows the relationship between the elements in a cXML request document and document definitions within WebSphere Partner Gateway:

**cXML element**
**document definition**

**cXML DOCTYPE**
Protocol

**DTD version**
Protocol version

**Request (type) For example, OrderRequest**
Document type

## Response

The target partner sends a cXML response to inform the source partner of the results of the cXML request. Because the results of some requests might not have any data, the `Response` element can optionally contain nothing but a `Status` element. A `Response` element can also contain any application-level data. During PunchOut, for example, the application-level data is contained in a `PunchOutSetupResponse` element. The typical `Response` elements are:

- ProfileResponse
- PunchOutSetupResponse
- GetPendingResponse

The following table shows the relationship between the elements in a cXML response document and document definitions within WebSphere Partner Gateway:

**cXML element**
**document definition**

**cXML DOCTYPE**
Protocol

**DTD version**
Protocol version

**Response (type) For example, ProfileResponse**
document type

## Message

A cXML message contains the WebSphere Partner Gateway document type information in the cXML `Message` element. It can contain an optional `Status` element identical to that found in a `Response` element. It would be used in messages that are responses to request messages.

The content of the message is custom defined by the business needs of the user. The element directly below the `<Message>` element corresponds to the document type created in WebSphere Partner Gateway. In the following example, SubscriptionChangeMessage is the document type:

```
<Message>
<SubscriptionChangeMessage type="new">
           <Subscription>
                <InternalID>1234</InternalID>
                <Name xml:lang="en-US">Q2 Prices</Name>
                <Changetime>1999-03-12T18:39:09-08:00</Changetime>
                <SupplierID domain="DUNS">942888711</SupplierID>
```

```
            <Format version="2.1">CIF</Format>
        </Subscription>
</SubscriptionChangeMessage>
</Message>
```

The following table shows the relationship between the elements in a cXML message and the document definitions within WebSphere Partner Gateway:

**cXML element**
**document definition**

**cXML DOCTYPE**
Protocol

**DTD version**
Protocol version

**Message**
Document type

The easiest way to tell the difference between a one-way message and a Request-Response document is the presence of a `Message` element instead of a request or response element.

A message can have the following attributes:

- `deploymentMode`, which indicates whether the message is a test document or a production document. Allowed values are production (default) or test.
- `inReplyTo`, which specifies to which message this message responds. The contents of the `inReplyTo` attribute is the payloadID of a message that was received earlier. This would be used to construct a two-way transaction with many messages.

## Content-type headers and attached documents

All cXML documents must contain a Content-type header. For cXML documents without attachments, the following Content-type headers are used:

- Content-Type: text/xml
- Content-Type: application/xml

The cXML protocol supports attachment of external files through MIME. For example, buyers often need to clarify purchase orders with supporting memos, drawings, or faxes. One of the Content-type headers shown in the following list must be used in cXML documents that contain attachments:

- Content-Type: multipart/related; boundary=<*something_unique*>
- Content-Type: multipart/mixed; boundary=<*something_unique*>

The `boundary` element is any unique text that is used to separate the body from the payload portion of the MIME message. Refer to the cXML User Guide at www.cxml.org for more information.

## Valid cXML interactions

WebSphere Partner Gateway supports the following cXML document definition interactions:

- From external partner to internal partner: None/cXML to None/cXML with Pass Through and validation
- From internal partner to external partner:

– None/cXML to None/cXML with Pass Through and validation.
– None/XML to None/cXML with Pass Through, validation, and transformation.

# Creating document definitions

## About this task

Use the following process to create a new document definition for a cXML document.

**Note:** You must ensure that the correct version of cXML is defined before you create a cXML document definition. The default is Version 1.2.009.

1. Click **Hub Admin** > **Hub Configuration** > **Document Definition**.
2. Click **Create Document Definition**. The Create document definitions page is displayed.
3. Select **Document Type** for Document type.
4. Perform one of the following tasks, depending on the type of document:
   - For requests, enter the request type (for example, `OrderRequest`) **Name** field.
   - For responses, if the Response does not have any child tags other than `<Status>`, enter `Response`. Otherwise, enter the next tag name following `<Status>`. In the example that follows, you would enter `Response` for the first Response element and `Profile Response` for the second.

```
<cXML>
    <Response>
        <Status code="200" text="OK"/>
    </Response>
</cXML>
<cXML>
    <Response>
        <Status code="200" text="OK"/>
   </ProfileResponse>
    </Response>
</cXML>
```

5. Enter **1.0** for **Version**.

   The version number is for reference only. The actual protocol version is derived from the DTD version within the cXML document.
6. Enter an optional **Description**.
7. Select **Yes** for **Document level**.
8. Select **Enabled** for **Status**.
9. Select **Yes** for all **Visibility** attributes.
10. Click **Package: None** folder to expand the package selection options.
11. Select **Protocol: cXML (1.2.009): cXML**.
12. Click **Save**.

# Creating interactions

## About this task

After you create the document definition, set up an interaction for the cXML document.

To create interactions, use the following procedure.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. If the cXML document is the source, under **Source**, expand **Package: None** and **Protocol: cXML**, and select **Document Type:**<*document _flow*>. If the cXML document is the target, expand **Package: None** and **Protocol: cXML**, and select **Document Type:** <*document_flow*> in the **Target** column.
4. Expand the source or target column for the other half of the interaction (the document that will be converted to cXML or the document that will be transformed from cXML) and expand its package and protocol and select its document type.
5. Select **Pass Through** from the **Action** list at the bottom of the page. (**Pass Through** is the only valid option supported for cXML documents.)

## Custom XML document processing

This section describes how you can configure the hub to route XML documents that are not handled by one of the other built-in routing protocols.

*Custom XML* is a WebSphere Partner Gateway term that is used to refer to XML documents that are not handled by one of the built-in protocols.

The way that custom XML documents are identified is by a process of elimination. Based on the ordering of the fixed inbound workflow protocol parse steps, the hub attempts to match XML documents with each of the standard protocols before the protocol parse step that handles custom XML is called. The custom XML handler is called for any XML document that does not match one of the standard XML document types.

To process a custom XML document, the protocol parser has to extract information from the document. Your collection of XML formats, document protocol definitions, and document type definitions provides the custom XML protocol parser with information it needs to recognize and process a document using your configuration.

From a high level, this is how the custom XML protocol works:

1. The XML document is parsed to obtain any of these that exist: value of the document DTD name, the root tag name space, and the root tag name.
2. Based on the identifiers obtained in the first step, a set of document families that contain XML formats are identified as a possible match for the document. You will learn how to create document families and XML formats later in "Creating XML formats" on page 149.
3. Each possible matching XML format from the families is applied to the document to see if it matches the document. Matching is discussed later in this section.
4. When a matching XML format is found, it is used to extract the data from the document that the hub uses to process the document. The document family that the matching XML format is a member of determines the document protocol used for routing. The matching XML format itself determines with the document type used for routing.

Using the Manage XML Protocols page, you can create document families that are associated with document protocols. Then you can populate the format families with XML formats that are associated with document types.

An XML format comprises two types of information:

- XPath expressions that are used to extract information from XML documents.
- Literal data that is used as a constant value.

XML formats are used by the Document Manager to retrieve the values that uniquely identify an incoming document and access information within the document necessary for proper routing and processing.

Setting up custom XML routing is a multi-step process. To do so, you must complete the following:

1. Create a protocol that will be used to route a set of related documents and associate it with a package or packages.
2. Create a document type for the format and associate it with the newly created protocol.
3. Create a document family to hold a set of XML formats that match documents that are to be routed with the protocol.
4. Add XML formats to the family that are each associated with one of the document types for the family protocol.

You then create interactions between the new document types so connections can be made.

These steps are described in the sections that follow. You can also find an example of these steps in "Setting up the hub for custom XML documents" on page 312.

## Creating XML formats

XML formats are used to identify and extract data from custom XML documents so they can be processed. XML formats are contained within document families. A document family is a collection of related XML formats that share a common DTD name, root element tag, or root element namespace. Therefore, there are three types of document families: DTD families, Root tag families, and Namespace families.

Document families serve two roles:

- They can determine how documents are routed. At runtime, when a document matches an XML format, the routing protocol and version that are associated with the format's family are used to route a document.
- They can help you to organize the XML formats in the system. When you are configuring the system, you can organize your XML formats by families. For example, you may group purchasing messages in a family named Purchasing messages, and you can then search for a document family to access the formats that are in a particular family.

### Creating a document family

#### About this task

To group related XML formats in a family, you must first create a family. To create a document family, complete the following:

1. Click **Hub Admin** > **Hub Configuration** > **XML Formats**.
2. Click **Create Document Family**.
3. In New document family view, enter a **Family name**.

   **Note:** More than one family can have the same identifier or name. The identifier type combined with the name forms a unique family key. For

example, suppose you want to route SOAP messages using the custom XML handler. If you have several different kinds of SOAP messages, you can classify them in families with different names all having Envelope as the root tag identifier.

4. Select a **Protocol** from the list of available protocols in the system. You should define a custom protocol before you define the family that uses it. You cannot change the protocol for a family after the family is created, so be sure to plan ahead.
5. Select a **Large file option**: None, Use large file processor, or Use namespace-aware large file processor.

   **None** means that the XML formats in the family can use XPath Version 1.0 expressions, but the size of the files that can be processed will be limited by several factors, including the document manager memory configuration, document manager workload, and the structure of the documents that are processed.

   **Use large file processor** or **Use namespace-aware large file processor** means that the file size is not a limitation but you are limited to using simple element path expressions in the XML formats that are members of the family.

   Use a large file option if you are writing XML formats that will match large documents that cannot be handled using the full XPath processor. If you select the namespace-aware option, element paths will include namespace prefixes when they appear in a document.
6. Select a document **Family type** from the list: DTD, Root tag, or Namespace.
7. Enter a **Family identifier** for the type of family you are creating:

*Table 23. Identifiers for family types*

| For this type of family | Enter this as the identifier |
|---|---|
| DTD | The DTD name |
| Root Tag | The root tag of messages that are in that family<br>**Note:** Omit the namespace prefix if there is one. |
| Namespace | The namespace of the root tag |

   This identifier is used at runtime to select a family of XML formats, one of which may be matched to the document and used to extract processing information from it. Note that if there are several families that use the same identifier, the formats in all of the families will be checked against the message until a match is found.
8. Click **Save** to save the new family or click **Cancel** to stop creating a document family or **Return** to return to the initial view.

## Finding a document family

### About this task

To view a document family, you must first find it. To find a document family, complete the following:

### Procedure

1. Click **Hub Admin** > **Hub Configuration** > **XML Formats**.
2. Select the protocol of the document family you want to view.
3. Enter the family name, if known. You can use an asterisk (*) to perform a wildcard search
4. Select the family type: Any type, DTD, Namespace, or Root Tag.

5. Select the large file option: None, Use large file processor, or Use namespace-aware large file processor
6. Click **Search**. All document families that fit your search criteria appear below the Search button.
7. Click the **View Details** icon beside a document family to see its details.

## Editing a document family

### About this task

In the Document Family details window, you can edit the properties for a family. To do so, complete the following:

### Procedure

1. Click the pencil button in the family details view to display a Document Family edit view. Notice that the protocol cannot be changed in this view. That is because there may have been messages routed using formats in the family and it would make debug difficult if the protocol associated with the family was changed.
2. In the Document Family edit view, you can now change the family name, family type, and the family identifier.
3. When you have made your changes, click **Save** to save them. Click **Cancel** or the crossed out pencil button to return to the details view of the family without saving any changes.

## Adding a new XML format to a family

### About this task

Once you have created a document family, you can add new XML formats to that family. To do so, complete the following:

**Note:** In this section, the term XPath expression is often used. When an XML format uses a large file option, this term should be taken to mean an Element path expression, which is a simple path from the root of a document to an element that has a value.

1. Starting from the Document family details view, click **Create XML Format**. The XML format definition view is displayed. This page is divided into four sections under the headings **Document type definition**, **Document type definition criteria**, **Document attributes**, and **User-defined attributes**.
2. Complete the **Document type definition** section.

   In the Document type definition section there is a selection list with the document types that are contained in the protocol associated with the document family. Select a **Document type** from this list. When a document matches the XML format, the protocol associated with the document family and the document type associated with the format are used to route the document.
3. Complete the **Document type definition criteria** section.

   The **Document type definition criteria** section and the **Document attributes** section include fields where you enter values and element paths if you are using a large file option or enter XPath expressions, prefix namespaces, and return types if you are not.

   **Value** In this field, enter a value for the format identifier. This is a required field.

**Element path**

In this field, enter an element path. This is a required field. Note that element path applies only to formats that use a large file option.

**XPath expression**

In this field, enter either a valid XPath expression for the document that matches the format or a literal string value that is returned as a constant for every document. This is a required field. Note that XPath expressions are used only in formats that do not use a large file option.

**Prefix namespace field**

In this field, enter the definition of the last namespace prefix, if any, used in your XPath expression. This is entered in the form prefix=namespace qualifier. For example, if the last namespace prefix in your expression is SOAPENV and its qualifier is http://schemas.xmlsoap.org/soap/envelope/ then you would enter SOAPENV=http://schemas.xmlsoap.org/soap/envelope/ for the Namespace prefix. Note that formats that use a large file option do not have prefix namespace fields as part of their definition.

**Return type**

In this field, select either Constant, Text, or Element tag name from the selection list. Use Constant when you want to interpret the XPath expression field as a string literal for all documents. Use Text when you want to use the XPath evaluation engine to evaluate the expression in the context of the document. Use Element tag name when you want to obtain the element name for the first element returned by the XPath evaluation of the expression. Note that formats that use a large file option do not have Element tag name as a return type.

In the Document type definition criteria section, you enter values and XPath expressions. The values and expression evaluation results are compared when documents are processed to determine if an XML format matches a document. When a match is found between a document and format and when the source and target business identifiers can be found using the format, the document is routed using the protocol and document type that are named in the Document type definition section. See Table 24 on page 153 for details about the fields in this section.

*Table 24. Document type definition criteria fields*

| Field | Required/ Optional | Action |
|---|---|---|
| Format identifier | Required | Enter the XPath expression or element path that defines the path to the content within the XML documents that uniquely identifies the document. For example, if the root tag looked like this <PurchasingMessage type="Purchase Order"> for purchase orders and looked like this <PurchasingMessage type="Order Confirmation"> for confirmations, then the XPath expression /PurchasingMessage/@type would return the text 'Purchase Order' for some messages and 'Order Confirmation' for others. Two XML formats, one for orders and another for confirmations, would be written and the 'Value' field for orders would say 'Purchase Order' and the 'Value' field for confirmations would say 'Order Confirmation'. At runtime, the proper format can be located by the system because it will look for a format where the expression evaluation gives a result that matches the value. When the match is found, the routing Document Type that is associated with the format is used by the system. |
| Format version | Required | Enter the XPath expression or element path that defines the format version. The format version is evaluated in a similar manner to that used for the format identifier. When the expression for the version matches the version value in a format, then the format may be used if the identifier also matches. Note that if there is only one version of a document, you can enter '1' for the expression with a Constant return type and '1' for the value. This means that the version will always match and the identifier alone is used to determine a matching format. |

4. Complete the **Document attributes** section.

   In the **Document attributes** section, you enter values and XPath expressions as you did for the **Document type definition criteria** section. See Table 25 for details about the fields in this section.

*Table 25. Document attributes fields*

| Field | Required/ Optional | Action |
|---|---|---|
| Source business identifier | Required | Enter the XPath expression or element path that defines the path of the source business ID within the XML document. This is used to identify the source partner for routing purposes. Notice that this data must be found for the format to be used. |
| Target business identifier | Required | Enter the XPath expression or element path that defines the path of the target business ID within the XML document. This is used to identify the target partner for routing purposes. Notice that this data must be found for the format to be used. |
| Document identifier | Optional | Enter the XPath expression or element path that defines the path for the document ID number within the XML document. This value will be displayed in the document viewer. |

*Table 25. Document attributes fields (continued)*

| Field | Required/ Optional | Action |
|---|---|---|
| Document timestamp | Optional | Enter the XPath expression or element path that defines the path for the document creation time stamp within the XML document. This value will be displayed in the document viewer. |
| Duplicate check keys 1 - 5 | Optional | Enter the XPath expressions or element paths that define the paths used to identify whether a document is unique or is a duplicate. |
| Synchronous flag | Optional | Enter an XPath expression or element path that evaluates to *true* or *false* , indicating whether this document type requires a synchronous response or not. You can either enter an XPath expression that uses the document content to set the value, or enter the string literal true or false with a return type of Constant. The attribute BCGDocumentConstants.BCG_GET_SYNC_RESPONSE will be set in the BDO during Channel Parse processing if this field is set to true. |
| Validation root element | Optional | Enter the XPath expression that defines the root node of the content (payload) of an enveloped message within the XML document. WebSphere Partner Gateway will validate a document starting with this element. You need to specify an action that performs validation for this to work. This field does not occur in formats that use a large file option. |
| Related document ID | Optional | Enter the XPath expression or element path that provides the document identifier of a previously routed document that the current document is associated with. For example, an Order Confirmation is typically related to a Purchase Order. The Purchase Order document identifier value can be obtained using an XPath expression (see above). If the Order Confirmation includes the Purchase Order identifier, then it can be obtained using the related document ID expression. Doing so will link the documents in the document viewer. |
| Search fields 1- 10 | Optional | Enter XPath expressions or element paths that define the path to document content that you want to use for custom searches within the XML document. In the Document Viewer, you can search for documents based on the values in these fields. |

5. Complete the **User-defined attributes** section.

   In the **User-defined attributes** section You can add custom user-defined attributes. You add an attribute by typing its name in the entry field and clicking **Add**. You then define this new attribute as you would for the other standard attributes by entering, as appropriate, the XPath expression, element path, prefix namespace, and by selecting a return type for this attribute

   Once you have added attributes, they are used in the same way that the standard attributes are used. If you want to remove a user-defined attribute from a format, click the red X that appears next to its name. User-defined attributes are intended for use by user-written handlers that process the document. The attribute names and their values are added to the business document when the document is processed. Your handler code can access these

by getting them from the business document using the names that you defined. See the *WebSphere Partner Gateway Programmer Guide* for more information.

6. After entering the values on this view, scroll to the bottom, and click **Save** to save the changes. Click **Cancel** or the crossed pencil button to cancel the changes and return to the family summary view.

## Routing XML messages with different namespace prefixes

### About this task

When routing XML messages, you must setup an XML Format definition that contains the exact namespace and prefix defined in the XML message. When you use different namespace prefixes, configure the routing of XML messages in the Console of WebSphere Partner Gateway. The three methods by which you can perform the configuration are as follows:

- Creating Document Family and XML Format for every message that will use the different namespace prefixes.
- Creating one Document Family and XML Format for the local-name (Schema Root Tag).
- Creating one Document Family and XML Format, using a combination of local-name and namespace.

**Creating Document Family and XML Format for each message that will be using different namespace prefixes:**

1. Navigate to **HubAdmin** > **Hub Configuration** > **XML Formats**.
2. Click **Create Document Family** link.
3. In the **New document family** page, create a new document family of type *Namespace*.
4. Click **Save**.
5. Click **Create XML format** link. This XML format will be created under the newly created Document Family.
6. In the **XML format definition** page, define the XML Format for the namespace and prefix to be used by the message.
7. Repeat steps 2, 3, 4, 5 and 6 for every XML format that is defined for the namespace prefix. However, create a different Document family for each XML format.

**Create one Document Family and XML Format for the local-name (Schema Root Tag):**

1. In the **New document family** page, create a **Document Family** of type *Root Tag*.
2. Create the XML Format under the newly created Document Family. When defining the **XPath expression**, use the local-name (Root Tag) for the Format identifier (**Source business identifier** and **Target business identifier**).
3. Click **Save**.
4. Send the XML message that contains the different XML prefix namespaces.

   **Note:** The local-name for the XML schema can also be used to define other fields in the XML Format, for example, Search Fields. Search Fields can also be defined with mapping commands using the DIS Client or through a custom written User Exits.

**Creating one Document Family and XML Format, using a combination of local-name and namespace:**

1. In the **New document family** page, create a Document Family of type *NameSpace*

2. Click **Save** to save the newly created document family.
3. Create the XML Format under the newly created Document Family. Define the XML Format, using a combination of local name (Root Tag) and namespace. For example, **XPath expression for Source business identifier:** *//*[namespace-uri()='http://edi.mycompany.com/2007/types/transnet' and local-name()="purchaseOrder"]/* [namespace-uri()='http://edi.mycompany.com/2007/types/transnet' and local-name()="senderID"]* **XPath expression for Target business identifier:** *//*[namespace-uri()='http://edi.mycompany.com/2007/types/transnet' and local-name()="purchaseOrder"]/* [namespace-uri()='http://edi.mycompany.com/2007/types/transnet' and local-name()="receiverID"]*
4. Send the XML message that contains the different XML prefix namespaces.

   **Note:** The combination of local-name and namespace for the XML schema can also be used to define other fields in the XML Format, for example, Search Fields. Search Fields can also be defined with mapping commands using the DIS Client or through a custom written User Exits.

## Creating a protocol definition

### About this task

The following steps describe how to create a custom XML protocol definition format:

1. Click **Hub Admin > Hub Configuration > Document Definition > Create Document Definition**.
2. For **Document Definition type**, select **Protocol**.
3. For **Name**, enter an identifier for the document definition. For example, for a custom XML protocol, you might enter Custom XML. This field is required.
4. For **Version**, enter a value for the version of your protocol. Numeric or String values are permitted.
5. Enter an optional description of the protocol.
6. Set **Document level** to **No**, because you are defining a protocol, rather than a document type (which you will define in the next section).
7. Set **Status** to **Enabled**.
8. Set **Visibility** for this protocol. You will probably want it to be visible to all partners.
9. Select the packages in which this new protocol will be wrapped. For example, if you want this protocol to be associated with the AS, None, and Backend Integration packages, select **Package: AS**, **Package: None**, **Package: Backend Integration**.
10. Click **Save**.

## Creating a document type definition

### About this task

Next, use the Create Document Definition page again to create a document type.

1. Click **Hub Admin > Hub Configuration > Document Definition > Create Document Definition**.
2. For **Document Definition type**, select **Document Type**.
3. For **Name**, enter an identifier for the document definition. For example, you might enter `Purchase order` as a name for the document type. This field is required.

4. For **Version**, enter a value for the version of your document type. Numeric or String values are permitted.
5. Enter an optional description of the document type.
6. Set **Document level** to **Yes** (because you are defining a routing object that corresponds to an actual document).
7. Set **Status** to **Enabled**.
8. Set **Visibility** for this flow. You will probably want it to be visible to all partners.
9. Click the **Expand** icon to expand each package you selected in step 9 on page 156. Expand the folder and select the name of the protocol you created in the previous section (for example, Protocol: Custom XML).
10. Click **Save**.

If you used the example values, the Manage Document Definitions page now contains a document type of Purchase order and a protocol of Custom XML under the AS, None, and Backend Integration packages.

## Finishing the configuration

After the protocol definition is defined, you will be able to choose it as the routing protocol to use for an XML document family. After you add document types to the protocol, you will be able to assign them to XML format definitions that are in the document family. Messages that match a format in the family will be routed using the protocol associated with the family and the document type associated with the matching format.

Before you can define any channels that use the new definitions, you need to enable interactions between your new protocols and document types and other protocols and document types. You also need to enable the B2B capabilities of your partners to allow them to send and receive documents using the new protocol and document types.

## Validating custom XML file against an XSD file

After the basic setup of the Custom XML (document type definition, creation of XML Family and XML Format, B2B Capabilities and Connection) is done and the XML is ready to be routed with the simple action "Pass Through", perform the following steps to allow validation of XML before pass through:

1. In the **Connections** page, set *Custom XML Pass Through with Validation* as the new action.
2. Navigate to **Hub Admin** > **Hub Configuration** > **Document Definition** .
3. Click **Edit attribute values** icon (blue arrow) for the custom XML Document Type.
4. Select **Upload map**.
5. Select corresponding XSD file and click **Upload**.
6. Repeat steps 2-3.
7. Click **Add attributes** to add Document Definition Context Attributes.
8. Select **Validation Map** and click **Save**.
9. In **Account Admin** > **Connections** search for the connection.
10. Click **Attributes** on the **Source** side of the connection.
11. Expand the collapsed node icon (blue folder) for Document Type.

12. From the **Validation Map** drop down, select the XSD validation map and click **Save**.

If you need to upload a newer version of the XSD file, you need to remove the old one first. This can be performed in the **Hub Admin** > **Hub Configuration** > **Maps** > **Validation Maps** page. After you have uploaded the new map, repeat step 12, as the deletion of a map resets this connection attribute.

# Using validation maps

WebSphere Partner Gateway uses validation maps to validate the structure of certain documents. If you want to associate a validation map with a document, first make sure the validation map is available to WebSphere Partner Gateway, as described in "Adding validation maps." For managing validation maps, see *Hub administration tasks Chapter of WebSphere Partner Gateway Administrator Guide*.

## Adding validation maps

### About this task

An action can have an associated validation map to ensure that the destination partner or back-end system can parse the document. Note that a validation map only validates the *structure* of the document. It does not validate the contents of the message.

**Note:** Once you associate a validation map with a document definition, you cannot disassociate them.

To add a new validation map to the hub, use the following procedure.

1. Save the validation map file to the hub or to a location from which WebSphere Partner Gateway can read files.
2. Click **Hub Admin > Hub Configuration > Maps > Validation Maps**.
3. Click **Create**.
4. Type a description of the validation map.
5. Navigate to the schema file you want to use to validate documents and click **Open**.
6. Click **Save**.

## Associating maps with document definitions

### About this task

To associate a validation map with a document definition, use the following procedure.

1. Click **Hub Admin > Hub Configuration > Maps > Validation Maps**.
2. Click the **View details** icon next to the validation map you want to associate with the document definition.
3. Click the **Expand** icon next to a package to individually expand to the appropriate level (for example, **Action** for a RosettaNet document).
4. Select the document definition you want associated with the validation map.
5. Click **Save**.

## Using Transformation maps

Steps to use transformation maps, which are used to convert a document from one format to another.

### About this task

WebSphere Partner Gateway uses transformation maps to convert documents from one form to another, for example, to convert XML document to EDI.

Following are the steps to use the transformation maps:

1. Log in to WebSphere Partner Gateway administrative console.
2. Click **Wizards**.
3. In the EIF Import Wizard, **Browse** and specify the location of the .EIF file.
4. Click **Import**.
5. In the Import Summary page, click **Next**.
6. In the Review Transformation Maps and Modify Interactions to be created screen, select the transformation map, add an interaction and select the action for the created interaction.
7. Click **Finish**.

**Important:** If you download a transformation map from the console of WebSphere Partner Gateway, a file of size 0 KB is downloaded; this is a known issue. As a workaround, use DIS client to download or extract transformation maps.

## Viewing documents

### About this task

The Document Viewer displays information about the documents that make up a document type. You can display raw documents and associated document processing details and events using specific search criteria. This information is useful if you are trying to determine whether a document was successfully delivered or to determine the cause of a problem.

To display the Document Viewer complete the following:

1. Click **Viewers > Document Viewer**.
2. Select the appropriate search criteria.
3. Click **Search**.

See the *WebSphere Partner Gateway Administrator Guide* for information on using the Document Viewer.

## Configuring non-repudiation logging

You can configure non-repudiation logging of messages using attributes of the package, protocol, or document flow used for routing documents. The attribute is named Non-Repudiation Required, and it can have a value of either Yes or No. The attribute definition is defined at the routing object level, and it can be overridden by changing it at the B2B capability level or at the connection level.

## Configuring message store

You can configure the message store using attributes of the package, protocol, or document flow used for routing documents. The attribute is named Message Store Required, and it can have a value of either Yes or No. The attribute definition is defined at the routing object level, and it can be overridden by changing it at the B2B capability level or at the connection level.

# Chapter 10. Configuring EDI document flows

This chapter describes how to configure the document definitions and interactions for standard EDI interchanges. Also included in this chapter are descriptions of receiving and transforming XML and record-oriented-data (ROD) documents. This chapter covers the following topics.



It is also possible to have an EDI interchange passed through with no de-enveloping or transformation. The steps for creating interactions for this type of exchange are presented in "EDI documents with Pass Through action" on page 104.

**Note:** You should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Overview of EDI

EDI is a method of transmitting business information over a network between business associates who agree to follow approved national or industry standards in translating and exchanging information. WebSphere Partner Gateway provides de-enveloping, transformation, and enveloping for the following EDI standards:

- X12, a common EDI standard approved by the American National Standards Institute
- UN/EDIFACT (United Nations Electronic Data Interchange for Administration Commerce and Support)
- UCS (Uniform Communication Standard)

The following sections provide a brief overview of EDI interchanges that conform to the X12, EDIFACT, and UCS standards and of the transactions and groups that are contained within the interchanges. Also described are how XML and ROD documents and EDI interchanges are transformed.

## The EDI interchange structure

An EDI interchange contains one or more business transactions. In X12 and related standards, a business transaction is called a *transaction set*. In EDIFACT and related standards, a business transaction is called a *message*. This document generally uses the term *transaction* or *business transaction* to refer to an X12 or UCS transaction set or an EDIFACT message.

EDI interchanges are composed of *segments* which in turn contain *data elements*. Data elements represent things such as a name, quantity, date, or time. A segment is a group of related data elements. Segments are identified by a segment name or segment tag, which appears at the beginning of the segment. (Data elements are not identified by name but are delimited by special separator characters reserved for this purpose.)

In some cases, it is useful to distinguish the detail or data segments in a transaction from other segments that are used for administrative purposes. The administrative segments are called *control segments* in X12 and *service segments* in EDIFACT. The *envelope* segments that delimit the boundaries of an EDI interchange are one example of these control or service segments.

EDI interchanges can contain three levels of segments. At each level, there is a header segment at the beginning and a trailer segment at the end.

An interchange always has an interchange header segment and an interchange trailer segment.

An interchange can contain one or more groups. A group in turn contains one or more related transactions. The group level is optional in EDIFACT but is required in X12 and related standards. When groups are present, there is a group header and a group trailer segment for each group.

A group (or an interchange, where groups are not present) contains one or more transactions. Each transaction has a transaction set header and a transaction set trailer.

A transaction represents a business document, such as a purchase order. The contents of the business document are represented by the detail segments between the transaction set header segment and the transaction set trailer segment.

Each EDI standard provides its own method for displaying the data within an interchange. The following table lists the segments for each of the three supported EDI standards.

*Table 26. Segments for supported EDI standards*

| Standard segment | X12 | UCS | EDIFACT |
|---|---|---|---|
| Interchange start | ISA | BG | UNB |
| Interchange end | IEA | EG | UNZ |
| Group start | GS | GS | UNG |
| Group end | GE | GE | UNE |
| Transaction start | ST | ST | UNH |
| Transaction end | SE | SE | UNT |

Figure 22 shows an example of an X12 interchange and the segments that make up the interchange.

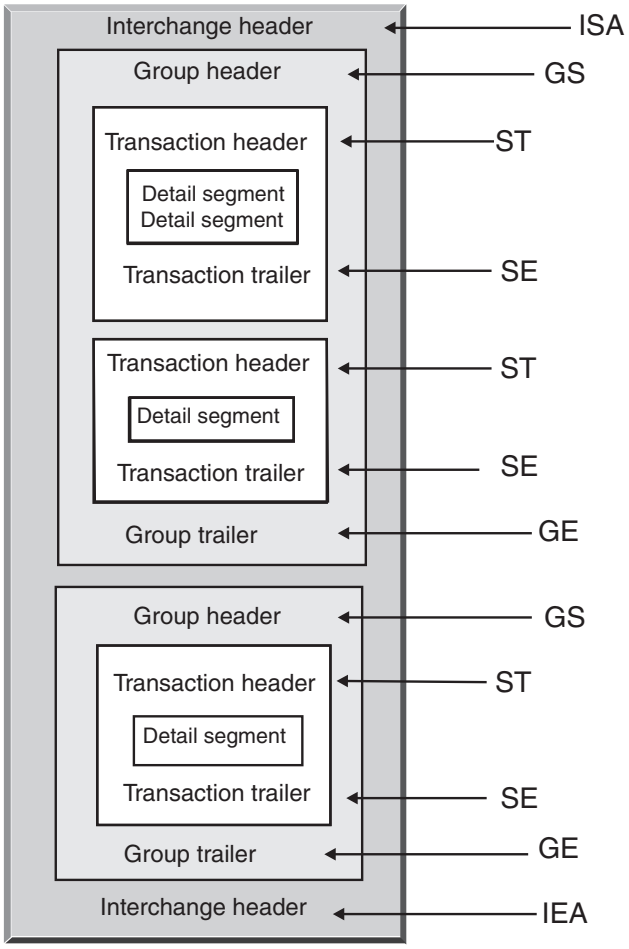



*Figure 22. An interchange envelope*

## Maps

The Data Interchange Services client mapping specialist creates transformation maps that describe how to change a document in one format to a document in a different format. You can, for example, have a transformation map that changes an X12 transaction into an EDIFACT message. You can also transform an EDI transaction into an XML document or a record-oriented data document.

Maps can be created using DIS or WTX design studio. DIS is used to create maps for WDI transformation, whereas WTX design studio is used for WTX transformation. The maps created using DIS cannot be migrated for WTX transformation, but must be rewritten. According to your action, the transformation engine will be selected if both are operational for you.

To create any map the definition of the source and target documents are required. The definitions of the source documents for EDI is supplied by WDI itself, but for ROD and XML you need to create using the DIS client. For this standard to be used by the runtime code it should be compiled. In the earlier versions, transformation maps are required for standard, but this version allows the compilation without the transformation map. The standard eif for EDI is imported, but for ROD it is created using DIS client. In case of XML the DTD/XSD is

imported into development database. For EDI, in the administrative console go to EDI wizards. The data formats / standards available in the EIF file will be displayed. You can import all at once or choose one or more to be imported. On successful selection the standard control string will be imported into runtime database.

The transformation map can also create multiple documents from a single document. This type of map makes use of *map chaining*, which produces multiple outputs from a single translation. In map chaining, after a source document has been successfully translated into a target document, a subsequent map is used to translate the source document again to produce another target document. This can be repeated as many times as needed to produce as many documents as needed.

In addition to transformation maps, you can use functional acknowledgment maps and validation maps. Functional acknowledgment maps provide instructions on how to produce a functional acknowledgment, which notifies the sender of an EDI document that the document has arrived. Several EDI Standard functional acknowledgment maps are installed when WebSphere Partner Gateway is installed. See"Setting up acknowledgments" on page 203 for a list of these maps.

When the sending hub expects a functional acknowledgement and it does not arrive within the time to acknowledge, the original document is resent. The number of retries and retry interval is configurable. This feature is not turned on by default. You need to manually set the value in the EDI properties. If the Time to acknowledgement is set to Yes, then values must be set for retry count and interval. The retry events are logged for the purpose of monitoring. If the retries exhausts without FA, then appropriate event will be logged for monitoring purpose.

Additional functional acknowledgment maps can be created by the Data Interchange Services client mapping specialist. WebSphere Partner Gateway generates a functional acknowledgement when an EDI transaction is validated and the EDI transaction has a functional acknowledgement map associated with it. The source document must be an EDI document.

WebSphere Partner Gateway provides a standard level of validation on the EDI document. If a functional acknowledgment is going to be generated, results from validation of an EDI document are saved. Validation maps are created to provide additional validation on an EDI document. The generation of a functional acknowledgment uses the functional acknowledgment map and the results from the validation of the EDI document. The functional acknowledgment map contains mapping commands that indicate how to use the validation results to create a specific functional acknowledgment. If a document is accepted for translation by the validation process, the appropriate data transformation map is used to translate the source document.

## Overview of XML and ROD documents

The Data Interchange Services client mapping specialist can create document definitions for XML and record-oriented data documents and then create transformation maps that change one type of document into another.

### XML documents

XML documents are defined by either an XML DTD or an XML schema. The Data Interchange Services client mapping specialist creates a transformation map based

on the DTD or schema that describes how to translate the XML document to another format. An XML document can be transformed into another XML document, a record-oriented data document, or an EDI transaction.

### ROD documents

The term record-oriented data (ROD) refers to documents that conform to a proprietary format. The Data Interchange Services client mapping specialist defines a ROD document definition, which refers to the way a business application structures data in a document. After a document definition is defined, the mapping specialist can create a map to transform the ROD document into another ROD document, an XML document, or an EDI transaction.

### Splitters and multiple documents

XML or ROD documents can enter the hub as individual documents or as a group of documents within the same file. Multiple documents might be put in the same file when, for example, a scheduled job at the partner or internal partner periodically uploads documents to be sent. If multiple XML or ROD documents arrive in one file, the Receiver calls the associated splitter handler (XMLSplitterHander or RODSplitterHandler) to split the set of documents. (The splitter handlers are configured when you create a target. See " Preprocess" on page 72 for information.) The documents are then reintroduced into the Document Manager to be processed individually.

**Note:** The sender and receiver IDs must be part of the ROD document definition associated with the transformation map. The information necessary to determine the document type and dictionary values must also be present in the document definition. Make sure that the Data Interchange Services client mapping specialist is aware of these requirements when creating the transformation map.

Multiple EDI interchanges can also be sent in one file. If multiple EDI interchanges arrive in one file, the Receiver calls the EDISplitterHandler to split the set of interchanges. The interchanges are then reintroduced into the Document Manager to be processed individually.

**Note:** Splitting is performed on the interchange, not on the individual transactions within the interchange. Transactions within the interchange are de-enveloped.

## Overview of creating document types and setting attributes

A document definition is made up of, at minimum, a package, a protocol, and a document type. The document definitions specify the types of documents that will be processed by WebSphere Partner Gateway.

Packaging refers to the logic that is required to package a document according to a specification, such as AS2. A protocol flow is the logic that is required to process a document that adheres to a certain protocol, such as EDI-X12. A document type describes what the document will look like.

The following sections briefly describe the overall steps for setting up a document flow between the internal partner and an external partner. The sections also describe the points at which you can set attributes.

## Step 1: Make sure the document definition is available

### About this task

Before you can send or receive a document, a document definition must be defined for the document. WebSphere Partner Gateway provides several default document definitions, including ones that represent functional acknowledgments. When you import transformation maps for EDI transactions or XML or ROD documents, the associated document definitions appear on the Document Definitions page. Similarly, if you import a functional acknowledgment map that is not already defined, the document definition for the acknowledgment appears on the Document Definitions page. You can also create your own document definitions.

As part of establishing the document definition, you can modify certain attributes. Attributes are used to perform various document-processing and routing functions, such as validation, checking for encryption, and retry count. The attributes you set at the document definition level provide a global setting for the associated package, protocol, or document type. The attributes that are available vary, depending on the document definition. Attributes for EDI document definitions have different attributes from RosettaNet document definitions.

For example, if you specify a value for **Allow a TA1 request** at the ISA document type level, the setting applies to all ISA documents. If you later set the **Allow a TA1 attribute** at the B2B capabilities level for a partner or the internal partner, that setting overrides the one set at the document definition level.

For attributes that can be set at multiple levels of the document definition, the values set at the document type level take precedence over those set at the protocol level, and the attributes set at the protocol level take precedence over the package level. For example, if you specify an envelope profile at the &X44TA1 protocol level but specify a different envelope profile at the TA1 document type level, the envelope profile you specify at the TA1 document type level is used.

You must have the document type listed on the Manage Document Definitions page before you can create interactions.

## Step 2: Create interactions

### About this task

You next set up interactions, which are templates for creating partner connections. Interactions convey how the document comes in, what processing is performed on the document, and how the document is sent from the hub.

For some protocols, you need only two flows, one to describe the document that is received into the hub (from the partner or internal partner) and one that describes the document that is sent from the hub (to the partner or internal partner). However, if the hub is sending or receiving an EDI interchange that will be de-enveloped into individual transactions or in which acknowledgments are required, you will actually create multiple interactions. For example, if you are receiving an EDI interchange at the hub, you will have an interaction that describes how the interchange is sent to the hub and how it is processed at the hub. You will also have an interaction for each transaction within the hub that describes how the transaction is processed. For EDI interchanges leaving the hub, you will have an interaction that describes how the interchange envelope is sent to the recipient.

## Step 3: Create partner profiles, destinations, and B2B capabilities

### About this task

Next, you create partner profiles for the internal partner and for external partners. You define destinations (which determine where documents will be sent) and B2B capabilities, which specify the documents the internal partner or a partner is capable of sending and receiving. The B2B capabilities page lists all the document types that have been defined.

You can set attributes at the B2B capabilities level. Any attributes you set at this level override those set at the document definition level. For example, if you set **Allow a TA1 request** to **No** at the document definition level for ISA documents but then set it to **Yes** at the B2B capabilities level, the value of **Yes** is used. Setting an attribute at the B2B level lets you tailor the attribute to a specific partner.

If you set the envelope profile at the protocol or document type level (on the Manage Document Definitions page) and then set it to a different value on the B2B Capabilities page, the latter value is used.

You must have the profiles and B2B capabilities of the internal partner and external partners defined before you can create connections between them.

## Step 4: Activate connections

### About this task

Finally, you activate connections between the internal partner and external partners. The connections that are available are based on the B2B capabilities of the partners and the interactions you created. The interactions depend on the document definitions being available.

For some exchanges, only one connection is required. For example, if a partner is sending a binary document to an internal partner back-end application, only one connection is needed. For the exchange of EDI interchanges in which the interchange is de-enveloped and the individual transactions are transformed, however, multiple connections are set up.

**Note:** For EDI interchanges that are being passed through as is, only one connection is required.

You can set attributes at the connection level. Any attributes you set at this level override those set at the B2B attributes level. For example, if you set **Allow a TA1 Request** to **Yes** at the B2B capabilities level but then set it to **No** at the connection level, the value of **No** is used. Setting a value for an attribute at the connection level lets you further tailor the attribute, depending on the routing requirements of the partners and applications involved.

# Overview of possible flows

This section gives you a brief overview of the types of transformations WebSphere Partner Gateway can perform. Details of these transformations and what you need to do to set them up are described in “Defining document exchanges” on page 191.

## EDI to EDI flow

WebSphere Partner Gateway can accept an EDI interchange from a partner or the internal partner, transform it into a different type of EDI interchange (for example, EDI-X12 to EDIFACT), and send the document to the internal partner or partner. The following steps occur when an EDI interchange is transformed into another EDI interchange:

1. The EDI interchange received at the hub is de-enveloped.
2. The individual transactions within the EDI interchange are transformed to the recipient's EDI format.
3. The transformed EDI transactions are enveloped and sent to the recipient.

Figure 23 shows an X12 interchange consisting of three transactions being de-enveloped. The transactions are transformed into EDIFACT format and are then enveloped and sent to the partner.

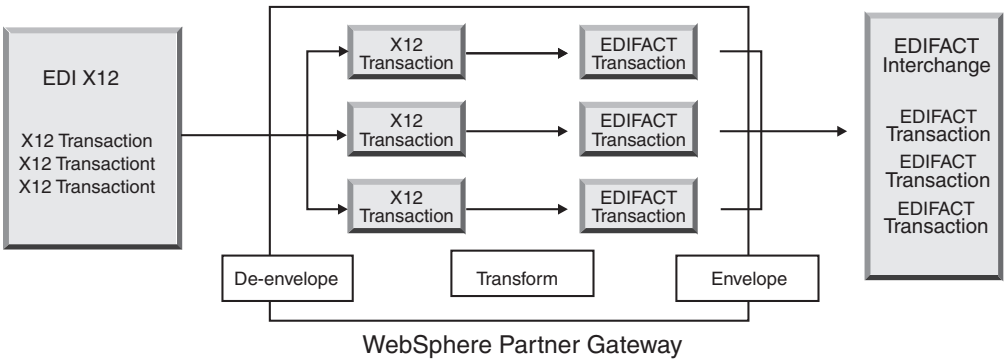



*Figure 23. EDI interchange to EDI interchange flow*

Each of the transactions has a transformation map associated with it, which specifies how the transaction is transformed. The transaction can be transformed into a single transaction or, if map chaining was used to create the map, multiple transactions. If Enveloper batching is turned on, transactions that enter the hub in one envelope will leave the hub in one envelope. However, if there are envelope breakpoints (for example, different values for EDI attributes or a different envelope profile) or if batching is turned off, the transactions will leave in different envelopes. See "Enveloper" on page 179 for a general description of the Enveloper (which is the component that gathers a set of transactions to be sent to a partner, wraps them in an envelope, and sends them). See "Batch mode" on page 180 for more information about batching.

The transaction might also have a validation map associated with it.

## EDI to XML or ROD flow

WebSphere Partner Gateway can accept an EDI interchange from a partner or the internal partner, de-envelope the interchange, and transform the resulting EDI transactions into XML or ROD documents.

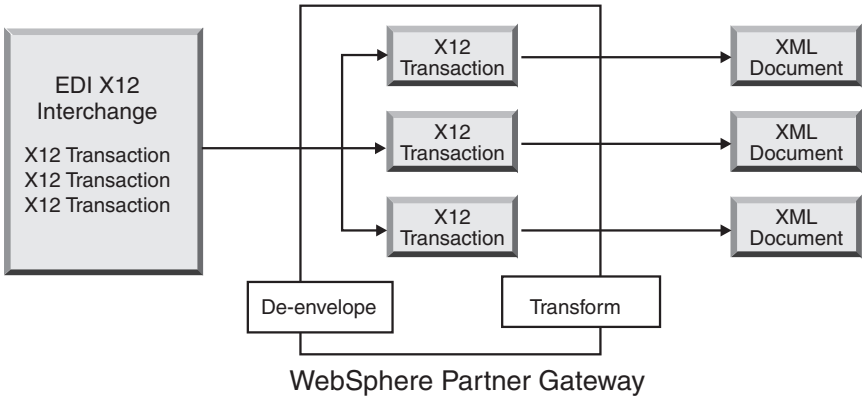



*Figure 24. EDI interchange to XML documents flow*

The transaction can be transformed into a single document or, if map chaining was used to create the map, multiple documents.

## XML or ROD to EDI flow

WebSphere Partner Gateway can receive XML or ROD documents from a partner or the internal partner, transform the documents into EDI transactions, envelope the transactions, and send them to the internal partner or a partner.

Figure 25 shows XML documents that are transformed into X12 transactions and then enveloped.

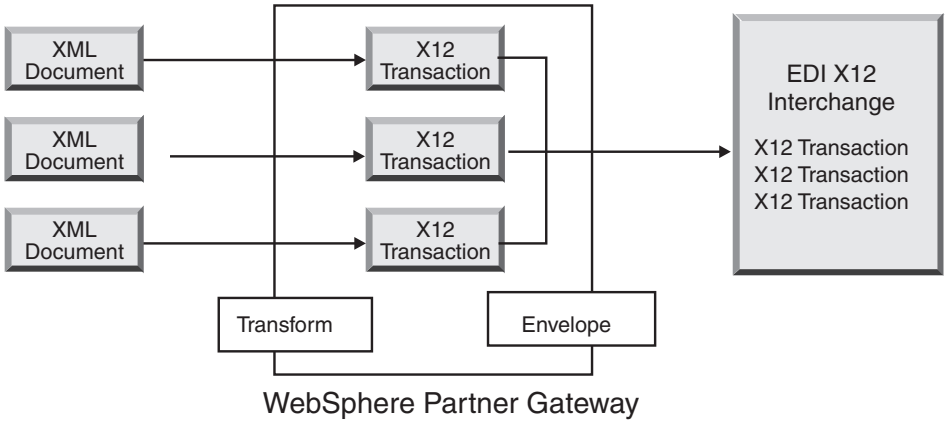



*Figure 25. XML document to EDI interchange flow*

One document can be transformed into multiple transactions (if map chaining was used to create the map), and the transactions can be enveloped into different interchanges. Figure 26 on page 170 shows an XML document that is transformed into three X12 transactions. Two of the transactions are enveloped together. One is put in a separate envelope.

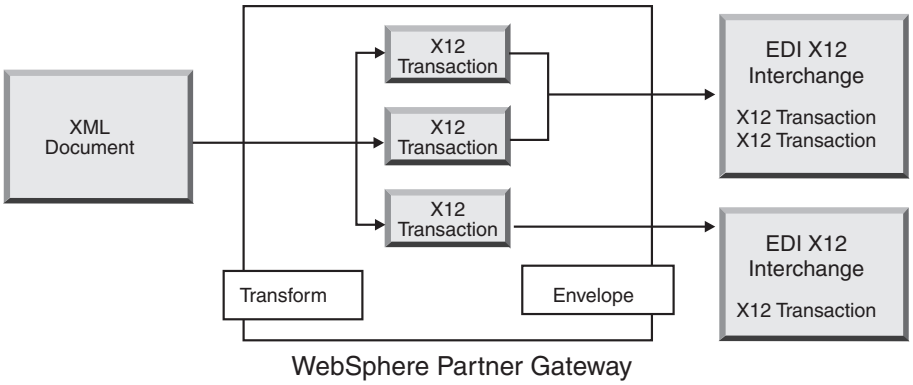



*Figure 26. XML document to multiple EDI transactions flow*

## Multiple XML or ROD documents to EDI interchange flow

WebSphere Partner Gateway can receive a file consisting of one or more XML or ROD documents from a partner or the internal partner, transform the document or documents into EDI transactions, envelope the EDI transactions into multiple envelopes, and send them to the internal partner or partner.

Each document can be transformed into a single transaction or, if map chaining was used to create the map, multiple transactions.

**Notes:**

1. Documents sent in a file must be of the same type--either XML documents or ROD documents, but not both.
2. ROD documents must be of the same type.

Figure 27 shows a set of XML documents being split, resulting in individual XML documents. The XML documents are transformed into X12 transactions, and the transactions are enveloped.

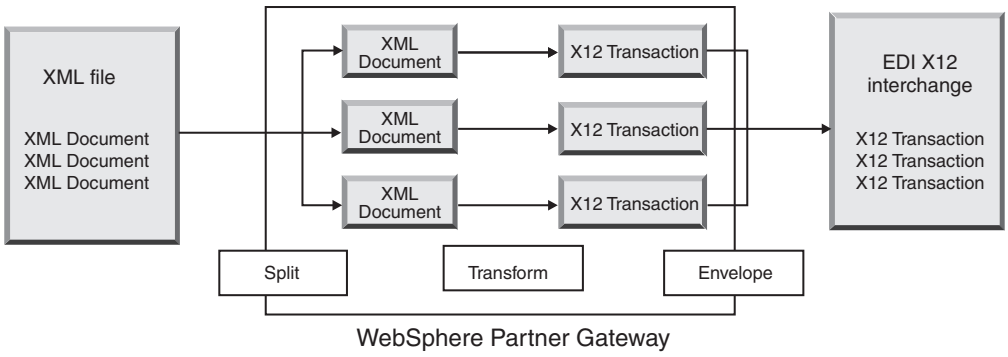



*Figure 27. Multiple XML documents to EDI interchange flow*

In Figure 27, the documents are split (by the XML Splitter Handler), and the transformed transactions are enveloped together. The XML Splitter Handler must have the BCG_BATCHDOCS option set to on (the default value) for this scenario to occur. If BCG_BATCHDOCS is set to on and the Enveloper batch mode is on, these transactions can be enveloped in the same EDI envelope. The Enveloper batch mode attribute is described in "Batch mode" on page 180.

## XML to ROD or ROD to XML flow

WebSphere Partner Gateway can receive an XML or ROD document from a partner or the internal partner, transform the document into the other type (XML to ROD or ROD to XML), and then send the document to the partner or internal partner.

Figure 28 shows a series of XML documents being transformed into ROD documents.

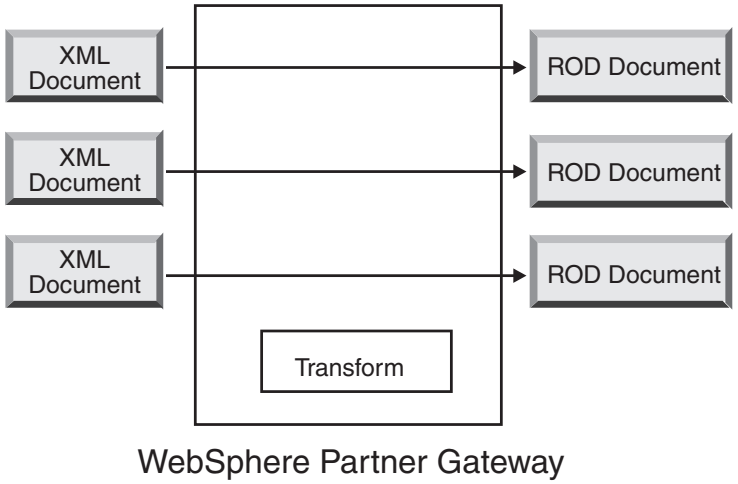



*Figure 28. XML document to ROD document flow*

The document can be transformed into a single document or, if map chaining was used to create the map, multiple documents.

## XML to XML or ROD to ROD flow

WebSphere Partner Gateway can receive an XML or ROD document from a partner or internal partner, transform it into a document of the same type (XML to XML or ROD to ROD), and then send the document to the partner or internal partner.

Figure 29 on page 172 shows XML documents that are transformed into XML documents of a different format.

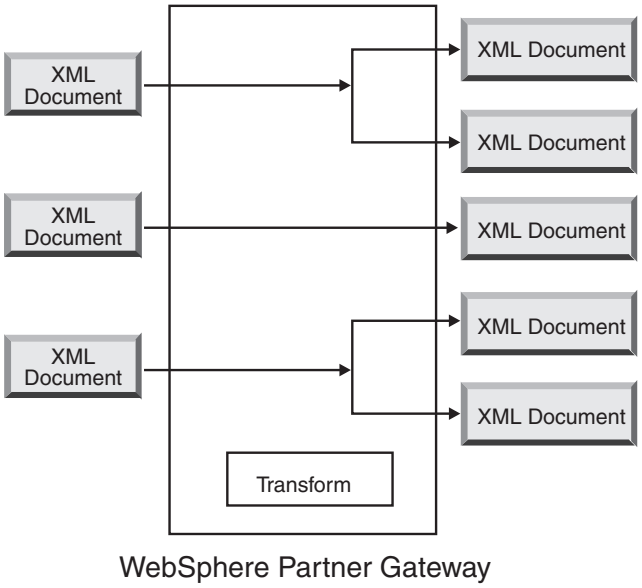


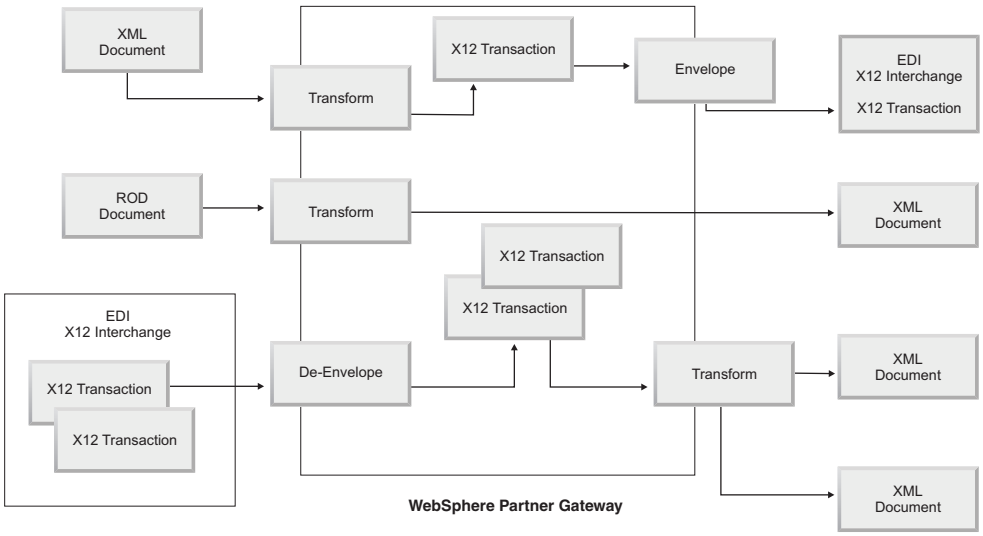

*Figure 29. XML document to XML document flow*

The document can be transformed into a single document or, if map chaining was used to create the map, multiple documents.

## Any to Any flow

WTX allows you to transform any to any format. WTX design studio is used for creating maps. The different flows are ROD to Any, XML to Any, and EDI to Any. Wherever required configure the splitter to split the documents. In case ROD is the source document, the routing information must also be set. XML formats provide the necessary routing information if XML is the source document. The different actions for the different flows are:

- ROD to Any - WTX transformation
- XML to Any - WTX transformation
- EDI to Any - EDI-Deenvelope if you want to deenvelope the interchange into transactions. Then actions EDI Reenveloper and WTX transformation are used for reenveloping the transactions and transforming them to EDI - Any format. EDI Validate if the transactions must be validated. Use EDI Interchange Validation if you want to validate the interchange without de-enveloping.




## Overview of transformation engines

WebSphere Partner Gateway supports two different transformation engines - native WDI and WTX.

**Native WDI** - Transformation maps are created in DIS client for native WDI. The various actions provided by WebSphere Partner Gateway for integration with WDI are EDI De-envelope, EDI Translate, EDI Validate, EDI Reenvelope, EDI Envelope, ROD Translate, and XML Translate. There is no separate configuration required for integration as it is native WDI.

**WTX** - Transformation maps are created using WTX design studio. The various actions provided by WebSphere Partner Gateway for integration with WTX are WTX Transformation, EDI Interchange Validation, EDI De-envelope, EDI Validate, EDI Reenvelope, and EDI envelope. RMI and native are two approaches for WTX. RMI is recommended in case WTX is not installed in the same machine as WebSphere Partner Gateway. The steps for invoking the WTX remotely is as follows:

1. In the DTXHome directory, open the rmiserver.properties file and modify the properties. For example, you can set the port number.
2. From DTXHome directory, run startrmiserver.bat.
3. In the Console Common properties, provide the hostname (where RMI server is running) and port number. Set the RMI server option to Yes.
4. Provide the physical location of the map.

For native approach, set System path as WTX Home directory. Also, set the property No for rmiuseserver.

## Envelope transactions from backend

When we use WTX in asynchronous case, the back end application consumes the EDI transactions generated by WTX and sends to WebSphere Partner Gateway for enveloping with backend packaging standard. The default backend headers are used to provide details of a transaction (x-aux-senderid, x-aux-receiverid, x-aux-protocol, x-aux-protocol-version, x-aux-process-type, x-aux-process-version, and x-aux-docSyntax). Backend package headers will contain information about

EDI-Dictionary/Protocol (say X12v4R1), Docsyntax ( EDI_transaction), and process transaction information (say 850) against the headers specified above. Refer to WTX envelope action section.

## How EDI interchanges are processed

An EDI interchange received at the hub is typically de-enveloped, and the individual transactions are processed. Often, standard EDI transactions (such as the X12 850 or the EDIFACT ORDERS, which represents a purchase order) are transformed into a form that can be understood by a back-end application. In addition, a functional acknowledgment is often sent to the partner to indicate that the interchange was received. The exchange of EDI interchanges, therefore, requires multiple actions ( EDI De-envelope, EDI Translate, EDI Validation, EDI Envelope, EDI Validate Interchange, EDI Renvelope, WTX Transformation, and WTX Envelope). For example, if the interchange contains two transactions and no acknowledgments are required, WebSphere Partner Gateway performs the following actions:

1. De-envelopes the interchange

   WebSphere Partner Gateway extracts information about the interchange from the envelope header and trailer segments at the interchange, group, and transaction levels. This information can include:

   - At the interchange level, the business identifiers of the sending and receiving partners, the usage indicator, which specifies whether the interchange is meant for a production or test environment, and the date and time the interchange was prepared
   - At the group level, the application identifiers of the sender and receiver and the date and time the group was prepared
   - At the transaction level, the type of transaction (such as X12 850 or EDIFACT ORDERS)
   - If validation is required for individual transactions, the EDI is de-enveloped. After the validation is over, the validated transactions are enveloped and sent either to transformation engine (WDI or WTX for processing) or to the destination depending on the action.
2. Transforms the first transaction according to the map associated with it.
3. Transforms the second transaction according to the map associated with it.
4. Delivers the transformed documents to the back-end application.

Similarly, when the hub sends a document or documents that originated at the internal partner back-end application, the documents are transformed into standard EDI transactions. The resulting EDI transactions are enveloped before being sent to the partner. As in the case of receiving an EDI interchange, multiple actions are required to create, envelope, and send an EDI interchange.

The individual transactions, groups, and interchanges are identified by control numbers. WebSphere Partner Gateway sets these numbers when an exchange takes place. You can customize the control numbers, however, as described in “Control numbers” on page 187.

The following illustration shows the overall picture of how an EDI interchange, packaged as AS, is sent from a partner, with the eventual goal of delivering two transformed XML documents to two different destinations on the internal partner back-end system. In this example, the 850 transactions are transformed into purchase orders that a back-end application can process. The 890 transactions are transformed in warehouse shipping orders that the back-end application can

process.

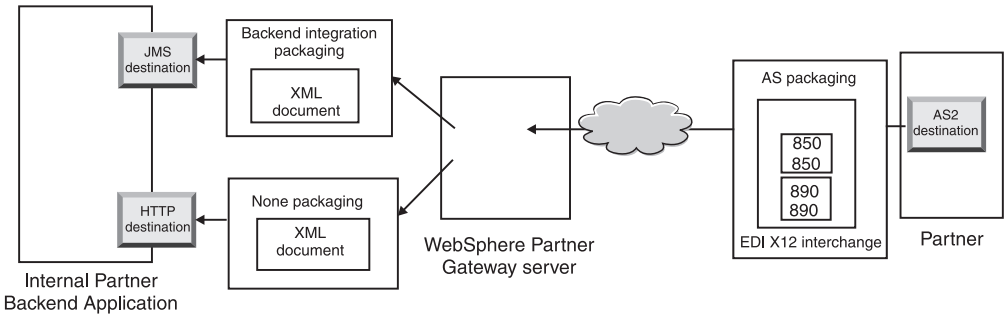



*Figure 30. Overall flow from a partner to the internal partner*

Instead of one connection from partner to internal partner, this exchange requires three connections:

- One from the partner to the hub to de-envelope the interchange. Because this is an intermediate step (the interchange is de-enveloped but is not delivered to the partner), the target side of the partner connection is N/A (not applicable).

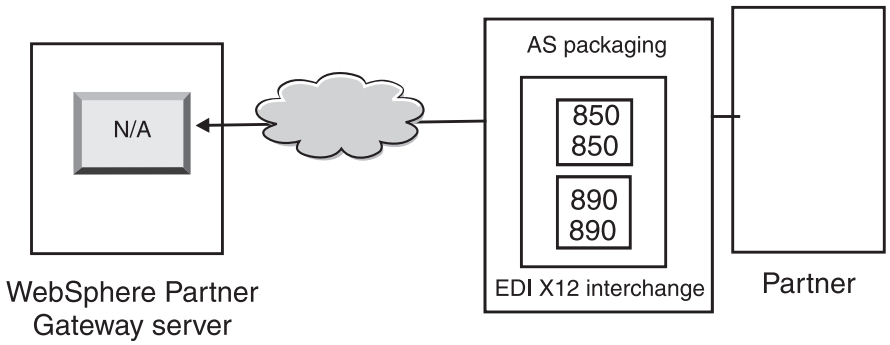



*Figure 31. The de-enveloping connection*

- One for the first transaction to be transformed and delivered to the JMS destination of the internal partner and one for the second transaction to be transformed and sent to the HTTP destination of the internal partner.

  For the transactions, the source packaging is not applicable because the transactions came in the original interchange that was de-enveloped by the system. Therefore, the source side of the transactions should have **Packaging: N/A** specified in the partner connection.

  For the transaction that is transformed into XML and that will flow to the back-end application over JMS, the target destination on the partner connection of this transaction should be specified as the JMS destination of the internal partner. For the transaction that was transformed into XML and that will flow to the back-end application over HTTP, the target destination on the partner connection of this transaction should be specified as the HTTP destination.

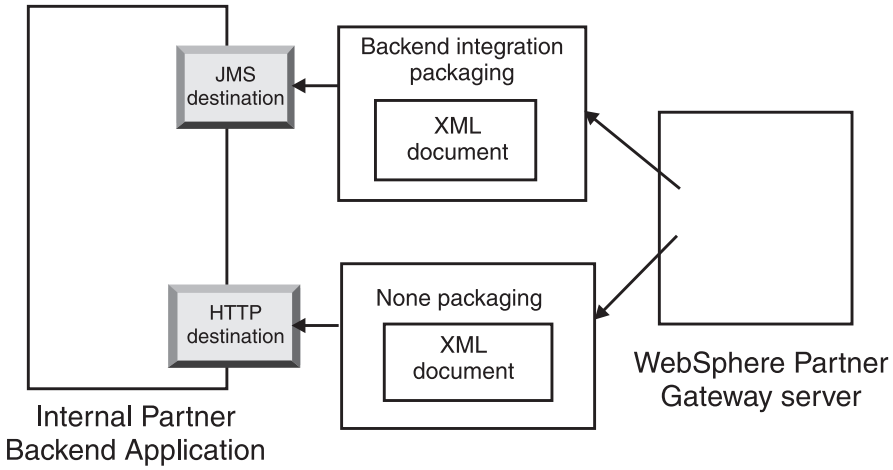



*Figure 32. Connections for individual transactions*

You can use the Document Viewer to view the interchange and the individual transactions, which, in the terms of the Document Viewer, are the *children* of the interchange. Using the Document Viewer, you can display the children associated with a source or target interchange, and you can display the events associated with them. The Document Viewer is described in the "Viewing Events and Documents" section of the *WebSphere Partner Gateway Administrator Guide*.

If the sender requests acknowledgments, you need additional connections:

- One for each of the acknowledgments sent back to the partner. The functional acknowledgments are generated by the system, and, therefore, the source side of the partner connection should have **Packaging: N/A** specified. Functional acknowledgments are enveloped before being delivered, and, therefore, the target side of the partner connection should also have **Packaging: N/A** specified. The Enveloper gathers these acknowledgments according to a schedule you set. See “Enveloper” on page 179 for information about setting the schedule.
- One to envelope the acknowledgments before they are sent back to the partner. The envelope is generated by the system, and, therefore, the source side of the partner connection should have **Packaging: NA** specified. The target side of the partner connection should have the target destination set to the destination of the partner and, in this case, with **Packaging: AS specified**. You can use a default envelope for the EDI standard, or you can customize envelopes. See “ Envelope profiles” on page 181 for information about customizing envelopes.

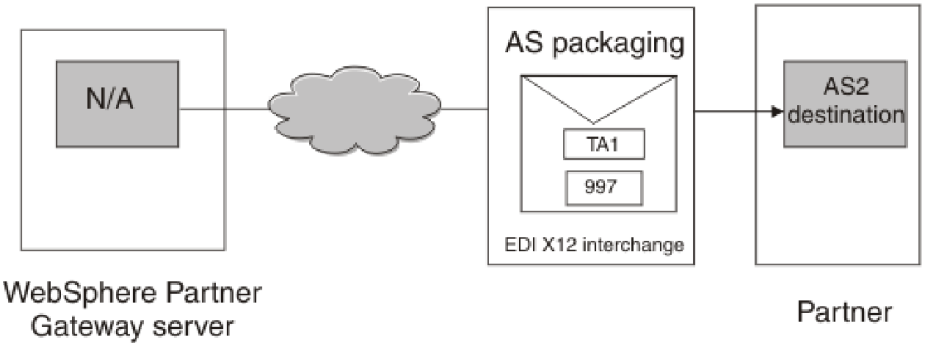



*Figure 33. Enveloping and sending acknowledgments to the originator*

### Synchronous transformation

WTX provides capability to transform Any format to Any format using a single map. An option is provided to directly make calls to WTX API for transformation. The de-nveloped and validated transaction is sent to WTX for processing after enveloping.

**Note:** See "Overview of EDI" on page 161for the various EDI formats available.

**One output** - the reroute attribute determines whether the output document should be reintroduced into the workflow or should be directly sent to the outbound workflow for processing.

**Multiple outputs** - based on the reroute flag, the child will be either passed directly to the outbound workflow or rerouted into the fixed inbound workflow for it to pass through a new channel.

### Asynchronous transformation

When an internal partner sends a message to external partner asynchronously, the external partner can make use of WESB/WMB or WTX for transformation. Configuration is not needed as WTX is considered as a JMS destination. WTX sends the document after processing to the back end and there is no information flow back to WebSphere partner gateway. The EDI document will be marked as Sent after successful delivery to JMS gateway.

## How XML or ROD documents are processed

An XML or ROD document is received at the hub as an individual document or as a group of documents in the same file. When a group of documents in the same file is received at the hub, WebSphere Partner Gateway performs the following actions:

1. Splits the set of documents into individual documents.
2. Transforms each document according to the map associated with it.
3. If the documents are transformed into EDI transactions, it envelopes the transactions and delivers them to the back-end application. If the documents are transformed into XML or ROD documents, it delivers the transformed documents to the back-end application.

If the XML or ROD document arrives as a single document, WebSphere Partner Gateway performs the following actions:

1. Transforms the document according to the map associated with it.
2. If the document is transformed into an EDI transaction, envelopes the transaction and delivers it to the back-end application. If the document is transformed into another XML or ROD document, the document is delivered to the back-end application.

Similarly, when the hub sends a document or documents that originated at the internal partner back-end application, the documents are transformed into XML or ROD documents, or they are transformed into EDI transactions. For EDI transactions, the transactions are enveloped before being sent to the partner. As in the case of receiving an EDI interchange, multiple actions are required to transform the document or documents, envelope the resulting transactions, and send the EDI interchange.

## Enveloping WTX integration and Polymorphic map

In WebSphere Partner Gateway metadata type tree is defined. You can configure and give information about the kind of data in each of the card. Typically the following properties are expected to be configured. The property names and values are case sensitive. Only boolean values are not case sensitive.

*Table 27. Metadata type tree properties*

| Property Name | Property Value | Description |
|---|---|---|
| BCG_DOCSYNTAX | EDI_INTERCHANGE<br>EDI_TRANSACTION<br>XML ROD | EDI_INTERCHANGE must be set if the output is an enveloped EDI Interchange. EDI_TRANSACTION must be set if the output is an EDI Transaction and is not enveloped. XML and ROD must be set for XML and ROD output accordingly. |
| BCG_REENVELOPE | true/false | If the value is true and BCG_DOCSYNTAX is EDI_INTERCHANGE, then the EDI envelope will be de-Enveloped. After De-Enveloping each transaction produced will be considered as an individual document for future steps. |
| BCG_REROUTE | true/false | If the value is true, the document will be re-routed. If false and the output is single, the existing BDO will be updated with the new file and sent out. |
| ProtocolName | As appropriate | The protocol name of the output Document. Mandatory in case ReRoute is set to true. This will be used to pick up the channel for rerouted document. |
| ProtocolVersion | As appropriate | The Protocol Version of the output Document. Mandatory in case ReRoute is set to true. This will be used to pick up the channel for rerouted document. |
| ProcessCode | As appropriate | The Process Code of the output Document. Mandatory in case ReRoute is set to true. This will be used to pick up the channel for rerouted document. |
| ProcessVersion | As appropriate | The Process Version of the output Document. Mandatory in case ReRoute is set to true. This will be used to pick up the channel for rerouted document. |
| SegmentCountElementName | SE01/UNT01 | If the output is EDI_TRANSACTION then we need to specify this attribute. This attribute should be set according to the kind of enveloping wanted. |
| SegmentCount | As appropriate | If the output is EDI_TRANSACTION then we need to specify this attribute. This attribute will have the information about the number of segments in the transaction. |

If the target is EDI after transformation, it must be enveloped before sending to external partners. The transformed output document can have any combination of formats. This depends on what is coded in the card number of the metadata card. This will contain the properties of other card details. The creator of the map will code the card. The different attributes that are considered are ReRoute, ReEnvelope, and DocSyntax. ReRoute and ReEnvelope can have True or False values, whereas DocSyntax can have any value entered by the user. Only if the value for DocSyntax is ediInchg, it will be considered for de-enveloping. The

following explains the possible outcome of the different combination of ReRoute and ReEnvelope values. It is assumed that the docSyntax set is to EDI_INTERCHANGE:

- ReRoute = True, ReEnvelope = False: the document is processed similar to any other document (XML or ROD).
- ReRoute = False, ReEnvelope = False: the document is processed similar to any other document (XML or ROD).
- Reroute = True, ReEnvelope = True: the document is DeEnveloped first. For each of the child transactions, a child bdo is created. The dictionary and document is set as protocol and process. Each of this ChildBDO (Transaction) is reRouted with N/A packaging. An appropriate channel must be present. The Enveloper profile can be configured in the target attributes of this channel. A separate channel for Envelope to flow through must be created.
- Reroute = False, Reenvelope = True: the document is DeEnveloped first. If single transaction is produced as output, the business document is updated with the transaction file as location and is sent out. If many transactions are produced as output, then child BDO's are created for no reroute and sent out. The target attribute of this channel is expected to be configured appropriately for Enveloper Profile. A channel for the Enveloper to flow must be present.

# Setting up the EDI environment

As mentioned in the previous section, you can specify many attributes that pertain to the exchange of EDI interchanges. For example, you can change the product-provided envelope profiles, you can define specific envelopes to be used for certain connections, you can set up control numbers that are assigned to the various parts of an interchange, and you can set connection profiles so that the same interchange can be delivered in a different way. These tasks are described in this section.

## Enveloper

The Enveloper is the component that gathers a set of transactions to be sent to a partner, wraps them in an envelope, and sends them. You schedule the Enveloper (or accept the default schedule) to indicate to WebSphere Partner Gateway when you want the Enveloper to look for transactions waiting to be sent. You can also update the default values for the lock time, queue age, and batch mode.

**Note:** Setting up the Enveloper is optional. If you do not change any of the values for the Enveloper, the product-provided default values are used.

### Locking

Each instance of the Document Manager has its own Enveloper. If you have two Document Managers installed on your system, you have two Envelopers. It is possible, therefore, for two (or more) instances of an Enveloper to attempt to poll transactions waiting to be enveloped. To ensure that a given transaction is polled by exactly one Enveloper, locks are used. Locks make sure that if multiple Envelopers are involved, only one Enveloper polls and processes a given transaction. Envelopers poll simultaneously but work on different transactions.

A time limit is set on the lock. The default value for an instance of the Enveloper to hold the lock is 240 seconds.

If the Enveloper has to wait for the lock, it is placed in a queue. The maximum queue age (the length of time the Enveloper should wait) is 740 seconds.

Typically, you will not need to change any of the default values for locking.

## Batch mode

Multiple documents that arrive in one file are split, according to the splitter handler you have set up for that type of document. (Configuring splitter handlers, which is part of defining targets, is described in " Modifying configuration points" on page 72.) One of the attributes of the splitter handler is BCG_BATCHDOCS. When BCG_BATCHDOCS is set to on (the default value), the splitter adds batch IDs to the documents after the documents are split.

The Enveloper has an attribute for batch mode, which is related to the BCG_BATCHDOCS attribute. If batch IDs were assigned to the individual documents, and if you accept the default value (on) for batch mode, the Enveloper makes sure that all documents that arrive together in the same file are processed before it envelopes and sends them, to ensure that the transactions are enveloped together. For example, suppose five XML documents arrive in the same file. The XML documents are to be transformed into EDI transactions and are intended to be delivered to the same recipient. After only three of the documents have been transformed, the Enveloper begins its scheduled polling for transactions. If batch mode is selected, the Enveloper does not process (envelope) the three transactions that are ready. Instead, it waits until all five transactions have finished processing before it envelopes and sends them. The transactions are placed in the same envelope, unless the applicable EDI standard prevents this.

## Modifying the default values

### About this task

To modify any of the default values for the Enveloper, perform the following steps:

1. Click **Hub Admin > Hub Configuration > EDI > Enveloper**.
2. Click the **Edit** icon.
3. Enter new values for **Maximum Lock Time (Seconds)** and **Maximum Queue Age (Seconds)** if you want more or less time assigned to these attributes.

   **Note:** Typically, you will not need to change any of the default values.
4. If you want to turn off batch mode, remove the check next to **Use Batch Mode**.
5. If you want to change how often the Enveloper checks for transactions waiting to be sent, perform one of the following sets of tasks:
   - To use interval-based scheduling (which is the default) but change the amount of time, enter a new time next to **Interval**. For example, if you change the value to 30 seconds, the Enveloper will check for documents every 30 seconds, envelope those documents, and send them to the recipient.
   - To use calendar-based scheduling, perform the following tasks:
     a. Click **Calendar Based Scheduling**.
     b. Choose the type of scheduling (**Daily Schedule**, **Weekly Schedule**, or **Custom Schedule**).
        - If you select **Daily Schedule**, select the time of day (the hours and minutes) when the Enveloper should check for documents.
        - If you select **Weekly Schedule**, select one or more days of the week in addition to the time of day.
        - If you select **Custom Schedule**, select the time of day and then choose **Range** or **Selective Days** for the week and the month. With **Range**, you indicate the start date and the end date. (For example, you can click **Mon** and **Fri** if you want the Enveloper to check for documents

at a certain time on weekdays only.) With **Selective Days**, you choose the specific days of the week and month.

6. Click **Save**.

## Envelope profiles

An envelope profile determines values that are placed in specific elements of the envelope. You assign the envelope profile to EDI transactions in the document definition **Envelope Profile** attribute. WebSphere Partner Gateway provides a predefined envelope profile for each supported standard (X12, EDIFACT, or UCS). You can use these predefined envelopes directly, you can modify them, or you can copy them into new envelope profiles. The steps for modifying an envelope profile or creating one are described in "Modifying the default values" on page 182.

The Envelope profiles have one field for each element in the envelope standard. The profiles provide literal or constant data for building header or trailer segments for transaction sets, messages, functional groups, and interchanges. You supply only the values that need to be populated and for which a value is not provided by another source.

The field names are designed to make cross-referencing easy. For example, the field UNB03 is the third data element in the UNB segment.

As described in " Envelope attributes," attributes set anywhere else take precedence over the values you set in the envelope profile. Some of the attributes can be overridden in document definition-related attributes or maps.

### Envelope attributes

Envelope attributes can be set at several different points during the configuration process, and they can also be set in the transformation map associated with the documents. For example, the Data Interchange Services client mapping specialist can specify the CtlNumFlag property when defining a map. This property can also be set as part of the envelope profile (in the **Control Numbers by Transaction ID** field). Any attributes set in the transformation map override the related values set at the Community Console. For example, if CtlNumFlag is set in the transformation map as **N** (no) and you enter a value of **Y** (yes) in the **Control Numbers by Transaction ID** field, the value of **N** is the one that is used.

Other envelope profiles can be set at the protocol level (from the Manage Document Definitions page or from the B2B capabilities page associated with a partner), or they can be set as part of the connection. The order of precedence is outlined in the following list:

1. Properties set in the transformation map take precedence over the associated attributes set in the Community Console.
2. Attributes set at the connection level take precedence over those set at the B2B capabilities level.
3. Attributes set at the B2B capabilities level take precedence over those set at the document definition level.
4. Attributes set anywhere (either in the transformation map or at the document definition, B2B capabilities, or connection level) take precedence over the values set in the envelope profile.

For a list of transformation map properties and their associated Community Console attributes, see "Data Interchange Services client properties" on page 419.

## Modifying the default values

### About this task

" Envelope profile attributes" on page 409 provides a table showing the default values used for each EDI standard envelope attribute if you do not enter a value in the profile or if you do not create a profile. Make sure the envelope profiles you are using supply any mandatory elements that are not provided by the system at runtime.

To set up an envelope profile, perform the following steps:

1. Click **Hub Admin > Hub Configuration > EDI > Envelope Profile**.
2. Perform one of the following sets of steps:
   - Create an envelope
     a. Click **Create**.
     b. Type a name for the envelope profile. This is the name that will appear on the Envelope Profiles list.
     c. Optionally, type a **Description** of the profile.
     d. Click the **EDI Standard** to which the envelope pertains. For example, if you are exchanging documents that conform to the EDI-X12 standard, select **X12**.
   - Modify an envelope
     a. Select one of the existing envelope profiles by clicking the **View details** icon next to the name of the profile.
     b. Click the **Edit** icon.
3. The **General** button is chosen by default. You can enter a value for any field except ENVTYPE, which is prefilled with the standard you chose in step 2d.

   You can add values for the following fields:

   - **Interchange Control Number Length**, to indicate how many characters should be used when a control number is assigned to an interchange within the envelope.
   - **Group Control Number Length**, to indicate how many characters should be used when a control number is assigned to a group within the envelope.
   - **Transaction Control Number Length**, to indicate how many characters should be used when a control number is assigned to a transaction within the envelope.
   - **Max Transactions Number**, to indicate the maximum number of transactions allowed in this envelope.
   - **Control Numbers by Transaction ID**, to indicate whether you want to use the transaction ID (as part of the key) when the set numbers are looked up in the database. If so, separate sets of control numbers are used per each transaction ID.

   The fields for the General envelope profile are the same across all three standards, except that EDIFACT has an additional field: **Create Groups for EDI**.

   If you have made any changes to the General page, click **Save**.
4. To specify values for the interchange, click **Interchange**. A new set of fields is displayed on the page. The fields vary, depending on the EDI standard. Note that some of the values are already filled in or will be filled in at run time.
   - For the EDI-X12 standard, you can change the following fields:

- **ISA01: Authorization Information Qualifier**, which is a code for the type of information in ISA02.
- **ISA02: Authorization Information**, which is information used to further identify or authorize the sender of the interchange data.
- **ISA03: Security Information Qualifier**, which is a code for the type of information in ISA04. Valid values are:

  **00** ISA04 is not meaningful

  **01** ISA04 contains a password
- **ISA04: Security Information**, which is security information about the sender or interchange data. The code in ISA03 defines the type of information.
- **ISA11: Interchange Standards ID**, which is a code for the agency that controls the interchange. Valid values are: **U** (US EDI community of ASC X12), **TDCC**, and **UCS**.

  **Note:** This attribute is used for X12 versions through 4010. In X12 4020, the ISA11 element is used for the repetition separator.
- **ISA12: Interchange Version ID**, which is the version number of the syntax used in the interchange and functional group control segments.
- **ISA14: Acknowledge Requested**, which is the sender's code for requesting an acknowledgment. Valid values are:

  **0** Request no acknowledgment

  **1** Request an acknowledgment that ISA and IEA segments were received and recognized
- **ISA15: Test Indicator**, which is an indication that the interchange is for testing or production. Valid values are:

  **T** For test data

  **P** For production data

- For the UCS standard, you can change the following fields:
  - **BG01: Communications ID**, which is the identification of the transmitting company.
  - **BG02: Communications Password**, which is a password the receiver assigns, to be used as agreed upon by the partners.
- For the EDIFACT standard, you can change the following fields:
  - **UNB0101: Syntax Id**, which is the identification of the agency controlling the syntax being used. The controlling agency is UNO. The level is A or B.
  - **UNB0102: Syntax Version**, which is the version number of the syntax identified by the Syntax ID.
  - **UNB0601: Recipients Reference/Password**, which is a password assigned by the recipient, to be used as agreed upon by the partners.
  - **UNB0602: Recipients Reference/Password Qualifier**, which is a qualifier to the recipient's password, to be used as agreed upon by the partners.
  - **UNB07: Application Reference**, which is the sender's identification of the functional area to which the interchange messages relate.
  - **UNB08: Priority**, which is the sender's code for processing priority, as agreed upon with the partner. Code A is the highest priority.
  - **UNB09: Acknowledgement Request**, which is the sender's code for requesting an acknowledgment.

- **UNB10: Communications Agreement Id**, which is the name or code for the type of agreement used for this interchange, as agreed to with the partner.
- **UNB11: Test indicator (Usage Indicator)**, which is an indication that the interchange is for testing. 1 indicates a test interchange.

If you have made any changes to the Interchange page, click **Save**.

5. To specify values for the groups within the interchange, click **Group**. A new set of fields is displayed. The fields vary, depending on the EDI standard.

   The fields on this page generally define the sender and receiver of the group.

   - For the EDI-X12 and UCS standards, you can enter values in the following fields:
     - **GS01: Functional Group ID**, which is an identification of the type of transaction sets in the group.
     - **GS02: Application Sender**, which is the name or code for a specific department in the sender's company.
     - **GS03: Application Receiver**, which is the name or code for the specific department in the receiver's company that is to receive the group.
     - **GS07: Group Agency**, which is a code used with GS08 to identify the agency that has control of the standard.
     - **GS08: Group Version**, which is a code for the version, release, and industry of the standard.
   - For the EDIFACT standard, you can enter values in the following fields:
     - **UNG01: Function Group ID**, which is an identification of the type of messages in the group.
     - **UNG0201: Application Sender ID**, which is the name or code for a specific department in the sender's company.
     - **UNG0202: Application Sender Id Qualifier**, which is the qualifier for the sender ID code. Refer to the data element directory for a list of code qualifiers.
     - **UNG0301: Application Receiver Id**, which is the name or code for the specific department in the recipient's company that is to receive the group.
     - **UNG0302: Application Receiver Id Qualifier**, which is the qualifier for the recipient ID code. Refer to the data element directory for a list of code qualifiers.
     - **UNG06: Controlling Agency**, the code that identifies the agency that has control of the message type in the functional group.
     - **UNG0701: Message Version**, which is the version number for the message type.
     - **UNG0702: Message Release**, which is the release number within the version number for the message type.
     - **UNG0703: Association Assigned**, which is the code, assigned by the responsible association, that further identifies the message type.
     - **UNG08: Application Password**, which is the password assigned by the specific department in the recipient's company.

   If you have made any changes to the Group page, click **Save**.

6. To specify values for the transactions within a group, click **Transaction** or, in the case of EDIFACT, **Message**. A new set of fields is displayed. The fields vary, depending on the EDI standard.

   - For the EDI-X12 or USC standard, you can enter a value for **ST03: Implementation Convention ID String**.

- For the EDIFACT standard, you can enter a value in the following fields:
  - **UNH0201: Message Type**, which is a code assigned by the controlling agency to identify the message type.
  - **UNH0202: Message Version**, which is the version number for the message type.
  - **UNH0203: Message Release**, which is the release number within the version number for the message type.
  - **UNH0204: Controlling Agency**, which is a code for the agency that has control of the message type.
  - **UNH0205: Association Assigned Code**, which is the code, assigned by the responsible association, that further identifies the message type.
  - **UNH03: Common Access Reference**, which is the key that relates all subsequent transfers of data to a common file. Partners can agree to using a key made up of components, but subelement separators cannot be used.

  If you have made any changes to the Transaction page, click **Save**.

7. Click **Save**.
8. Repeat steps 2 on page 182 through 7 for any other envelope profiles you want to define or change.

After an envelope profile is defined, it is listed on the Envelope Profiles list. From the list, you can select the profile and then click the **Where Used** icon to determine the connections using the profile.

## Connection profiles

You use connection profiles with de-enveloped transactions and with EDI interchanges created by the Enveloper. For transactions, the connection profile determines how the transaction is processed after it is de-enveloped. For interchanges, the connection profile determines how the interchange is delivered.

Use the Connection Profile window to create a new profile or to edit existing profile information. The name of each currently defined profile and its description, if any, are shown in the Connection Profiles List. See the *WebSphere Partner Gateway Hub Configuration Guide* for more information about Connection Profiles.

### Transactions

When an EDI Interchange comes into WebSphere Partner Gateway, the first action is typically to de-envelope the interchange into the individual transactions. When the transactions are created, the De-envelope action sets the **Interchange usage indicator** and group information (**Group application sender identifier**, **Group application receiver identifier**, and **Group application password**) in the transaction metadata. Each transaction is then re-processed by WebSphere Partner Gateway in its own workflow.

Suppose you have two transactions of the same type (for example, 850) that need to be handled differently, depending on the group they were in or the values of their Interchange usage indicators. If the **Usage Indicator** is Production (**P**), for example, you might want one map (A) to be used, and if the **Usage Indicator** is Test (**T**), you want a second map (B) to be used. Two similar connections are required for this 850 transaction, with the only difference being that one connection uses map A and the other connection uses map B.

Because the transactions are otherwise the same (they have the same source and target partner, package, protocol, and document type), the Document Manager

needs a way to determine which connection to use. It does this by matching the connection profile attribute you set to the transaction metadata. In this example, if you create two connection profiles -- one (CPProduction) with the **EDI Usage Type** set to **P** and the other (CPTest) with the **EDI Usage Type** set to **T**, the Document Manager matches the transaction with the Usage Indicator of P with the CPProduction profile. It then knows to use map A to translate the transaction.

The example in this section used the **Interchange usage indicator** attribute, but you can also use the **Group sender application identifier**, **Group receiver application identifier**, and **Group application password attributes** as the distinguishing factor for a transaction.

## Interchanges

For interchanges, you use the **Connection Profile Qualifier 1** attribute.

For example, suppose you are in the midst of migrating your company from using a VAN (None packaging) or the Internet (AS2 packaging). You want 840 (Request for Quote) transactions to use the VAN and 850 (Purchase Order) transactions to use the Internet. You set up two partner connections, both with the same source interchange but with different targets (one with None packaging and the other with AS2 packaging). The connection profiles help distinguish between the two connections.

Setting up the connection profile for interchanges involves several steps. These are the steps you would perform to create two connection profiles for the example:

1. Create two connections for the transactions. Set the **Connection Profile Qualifier 1** attribute on the "To" side of both connections. The value should be meaningful (for example, ConNone and ConAS2).
2. Define two connection profiles (for example, CPNone and CPAS2), each with the **Qualifier1** value set to match the **Connection Profile Qualifier1** attributes you set in step 1 (ConNone and ConAS2).
3. Create two connections for the interchange. Each connection has the same source packaging (N/A) but different target packaging (None and AS2). The partner connection with the connection profile CPNone will have the target destination set to the FTP Scripting destination that can connect to the VAN. The partner connection with the connection profile CPAS2 will have the target packaging set to AS.
4. Associate the appropriate connection profile with each one.

The Enveloper uses the **Connection Profile Qualifier 1** attribute on the "To" side of the partner connection as an envelope break point. Therefore, transactions having different values for the **Connection Profile Qualifier 1** attribute will be enveloped in different envelopes. When you set different values for the transactions, the Enveloper will never envelope the 840 and 850 transactions in the same interchange.

When the Document Manager looks up the connection, the two possible connections are found, but the one with the matching connection profile is used.

## Setting connection profiles

### About this task

Setting connection profiles is optional. If you have no need to have more than one connection for each type of document you will be exchanging for a partner, skip this section.

To set up a connection profile:

1. Click **Hub Admin > Hub Configuration > EDI > Connection Profiles**.
2. Click **Create Connection Profile**.
3. On the Connection Profile Details page, type a required name for this connection profile.
4. Type an optional description of the profile.

   The name and description (if you enter a description) will appear on the Connection Profile List page.
5. Optionally, enter a value for **Qualifier 1** to indicate the value that determines which connection to use for an EDI interchange. See " Interchanges" on page 186 for an example of using **Qualifier 1**.
6. Optionally, enter a value for **EDI Usage Type** to indicate whether this is a test, production, or information interchange. See " Transactions" on page 185 for an example of using **EDI Usage Type**.
7. Optionally, enter a value for **Application Sender ID** to indicate the application or company division associated with the sender of the group.
8. Optionally, enter a value for **Application Receiver ID** to indicate the application or company division associated with the recipient of the group.
9. Optionally, enter a value for **Password** if a password is required between the application sender and application receiver.
10. Click **Save**.

For those transactions that you want to put into certain interchange envelopes, you can specify the **Connection Profile Qualifier 1** attribute value that corresponds to the connection profile with the same value for attribute **Qualifier 1**. The **Connection Profile Qualifier 1** attribute can be set at the protocol level of a document definition (for example, you could edit the attributes of the X12V5R1 protocol on the Manage Document Definitions page to indicate which connection profile to use by clicking the corresponding **Connection Profile Qualifier 1** attribute value). Then when you activate the interchange connection, associate the connection profile by clicking the **Connection Profile** button and selecting the profile from the list.

## Control numbers

The Enveloper uses control numbers to provide unique numbering for interchanges, groups, and transactions within an envelope. Control numbers are established for the internal partner and for external partners. When the exchange of documents takes place, control numbers are also generated for the *pair* of partners.

For each partner that has EDI B2B Capabilities, there is a set of seed initialization values for control numbers. These values are used the first time an EDI interchange is created and sent between a partner pair. The initialization values apply to the partner to whom the interchange is sent. After a document has been sent from one partner to another, the last numbers used can be viewed in the Current Control Numbers page. There can be several entries for a given partner pair if **Control Numbers by Transaction Id** is set to **Y**. After an entry exists, it is used to generate new control numbers.

As part of control number initialization, you can use masks to modify the normal control number creation by the Enveloper. Masks are used to base the control number on either the interchange or group control number. The mask descriptions

follow. Replace the *n* in the edit mask with the number of bytes you want to use to create the control number value. See Table 28 for descriptions of the available codes:

*Table 28. Control number masks*

| Code | Control Number | Description |
|---|---|---|
| G | Transaction | The transaction control number is the same as the group control number. Only one transaction for each group is allowed. |
| G*n* | Transaction | *n* bytes are taken from the group control number. The remainder of the transaction control number is padded with zeros to its maximum size. Only one transaction for each group is allowed. |
| C | Group, Transaction | The remaining bytes in the group or transaction control number field are used to maintain a control number for this partner. |
| V | Group, Transaction | An incrementing value is used so that the first group or transaction has a value of 1, the second a value of 2, and so on. |
| V*n* | Transaction | An incrementing value *n* bytes long is used so that the first transaction has a value of 1, the second a value of 2, and so on. |
| G*n*C | Transaction | *n* bytes are taken from the group control number and the remaining bytes in the transaction control number field are used to maintain a control number. The number of positions left determines the maximum value of the control number. For example, G5C leaves four positions; therefore the maximum value is 9999. The control number cycles from the maximum value to 1. |
| G*n*V | Transaction | *n* bytes are taken from the group control number. For the remaining bytes in the transaction control number field, an incrementing value is used so that the first transaction has a value of 1, the second a value of 2, and so on. |
| G*n*V*m* | Transaction | *n* bytes are taken from the group control number. For the remaining bytes, up to *m* bytes in the transaction control number field, an incrementing value is used so that the first transaction has a value of 1, the second a value of 2, and so on. |
| I | Group, Transaction | The group or transaction control number should be the same as the interchange control number. Only one group is allowed for the interchange, and only one transaction is allowed for the group or interchange. |
| I*n* | Group, Transaction | *n* bytes are taken from the interchange control number. The remainder of the group or transaction control number field is padded with zeros to its maximum size. Only one group is allowed for each interchange, and only one transaction is allowed for each group. |

*Table 28. Control number masks (continued)*

| Code | Control Number | Description |
|---|---|---|
| I*n*C | Group, Transaction | *n* bytes are taken from the interchange control number. The remaining bytes in the group or transaction control number field are used to maintain a control number. The number of positions left determines the maximum value of the control number. For example, I5C leaves four positions; therefore the maximum value is 9999. The control number cycles from the maximum value to 1. |
| I*n*V | Group, Transaction | *n* bytes are taken from the interchange control number. For the remaining bytes in the group or transaction control number field, an incrementing value is used so that the first group or transaction has a value of 1, the second a value of 2, and so on. |
| I*n*V*m* | Transaction | *n* bytes are taken from the interchange control number. For the remaining bytes, up to *m* bytes in the transaction control number field, an incrementing value is used so that the first transaction has a value of 1, the second a value of 2, and so on. |
| I*n*G*m* | Transaction | *n* bytes are taken from the interchange control number, and a maximum of *m* bytes are taken from the group control number. If *n* plus *m* is greater than 9, only 9 - *n* bytes are taken from the group control number. For example, using I4G6, 4 bytes are taken from the interchange |
| I*n*G*m*C | Transaction | *n* bytes are taken from the interchange control number, and *m* bytes are taken from the group control number. The remaining bytes in the transaction control number field are used to maintain a control number. The number of positions left determines the maximum value of the control number. For example, I2G4C leaves three positions; therefore the maximum value is 999. The control number cycles from the maximum value to 1. |
| I*n*G*m*V | Transaction | *n* bytes are taken from the interchange control number, and *m* bytes are taken from the group control number. For the remaining bytes in the transaction control number field, an incrementing value is used so that the first transaction has a value of 1, the second a value of 2, and so on. |
| I*n*G*m*V*o* | Transaction | *n* bytes are taken from the interchange control number, and *m* bytes are taken from the group control number. For the remaining bytes, up to *0* bytes in the transaction control number field, an incrementing value is used so that the first transaction has a value of 1, the second a value of 2, and so on. |

# Control number initialization

## About this task

To configure control numbers that the Enveloper will use, perform the following steps:

1. Click **Hub Admin > Hub Configuration > EDI > Control Number Initialization**.
2. Type a partner's name and click **Search** or click **Search** without entering a name to display all partners. If you leave **EDI-capable** checked, you limit the search to those partners that have EDI document B2B capabilities. If you remove the check, you search all partners.
3. Click the **View details** icon next to the partner.
4. The partner's current control number assignments (if any) are listed on the Control Number Configuration Details page. Click the **Edit** icon to add or change the values.
5. Type (or change) the value next to **Interchange** to indicate the number you want to use to initialize control number generation for interchanges.
6. Type (or change) the value next to **Group** to indicate the number you want to use to initialize control number generation for groups. Alternatively, you can click **Mask** and type a mask to be used instead of a fixed value.
7. Type (or change) the value next to **Transaction** to indicate the number you want to use to initialize control number generation for transactions. Alternatively, you can click **Mask** and type a mask to be used instead of a fixed value.
8. Click **Save**.

## Current control numbers

For a given partner-pair that already has data in the control table, you can change the control number generation. You can:

- Reset the control number generation for the pair to an initial state.
- Edit the interchange, group, or transaction number (or any combination of these numbers) and save it with a new value.

**Note:** Resetting control number generation or editing a group or mask should be done with caution so that numbers out of sequence or duplicate control number problems do not occur. You might want to perform either of these actions during test phase or if a partner specifically requests different control numbers.

To determine which partners have control numbers assigned (and to determine what those numbers are), you use the Current Control Numbers feature.

1. Click **Hub Admin > Hub Configuration > EDI > Current Control Numbers**.
2. Perform one of the following sets of steps:
   - If you want to see the current status of all partners, leave **Any Partner** selected in the partner lists, and click **View Current Status**.
   - If you want to view the status of selected partners, perform the following steps:
     a. Enter the name of the source and target partners and click **Search**. If you want to limit the search results to only those partners who are exchanging EDI documents, leave **Find EDI-Capable** checked.
     b. From the resulting lists, select one or more partners from each list, and click **View Current Status**.

# Defining document exchanges

You can define document exchanges manually or using wizards. If you want to define your connections using the wizards, see "Defining document exchanges using wizards." If you want to do this manually or manually modify your connections, see "Defining document exchanges manually" on page 193.

## Defining document exchanges using wizards

WebSphere Partner Gateway includes two wizards to help you define document exchanges. These are the EIF Import wizard and the EDI Connection wizard.

The EIF Import wizard guides you though the steps needed to import maps contained within EIF files, displays the details of the uploaded maps, associates these maps with the correct Routing Objects, and creates logical Interactions. On completion of the wizard, the new maps are uploaded and any necessary interactions are created in the system. You should then use the EDI Connection Wizard to create connections using your newly uploaded maps.

**Note:** To avoid confusion, only one user can use the EIF Import wizard at a time.

The EDI Connection wizard can be used after the EIF wizard and guides you though the steps needed to configure an EDI interaction (sending or receiving an EDI document). On completion of the wizard, the selected partners are fully configured for the EDI interaction. This includes enabling B2B capabilities, creating Valid Interactions, creating Partner Connections, and assigning the necessary EDI attributes. The Connection Wizard generates suggested Partner Connections based on your inputs. The full list of possible generated connections is listed here:

- De-Enveloper for base message
- Transformation
- Enveloper for base message
- TA1 generation
- FA generation
- Enveloper for TA1 and/or FA
- De-Enveloper for TA1 and/or FA

Both of these wizards are located under the Wizards tab in the console.

### Importing maps using the EIF Import wizard

#### About this task

To import maps using the EIF Import wizard, complete the following steps:

1. Start the WebSphere Partner Gateway Console.
2. Click **Wizards**.
3. Click **EIF Import Wizard**.
4. Enter the name of the file you want to import, or click **Browse** to find it.

   **Note:** When importing an EIF file containing multiple maps, ensure that the map names contained in the file are unique. If multiple maps are uploaded in the same EIF file with the same map name, the last matching map overwrites the previous matching maps in the database.
5. Click **Import**.

6. A list of the maps that have been successfully imported displays. Click **Finish** to accept the default values or click **Next** to view or modify them.
7. If you clicked **Next**, you are then asked to review the transformation maps and modify any interactions. Select a transformation map. If an Interaction exists, it displays as read-only. To add an interaction, click **Add an Interaction**.
8. On the Add an Interaction window, select an interaction and click **Add this Interaction** to add an Interaction to the list.
9. When you finish reviewing the transformation maps, click **Next** to review the validation maps.
10. Review the imported validation maps. If they are okay, click **Finish**. If you want to view the FA maps, click **Next**.
11. Review the imported FA maps, and click **Finish** and a final window displays showing the maps that have been successfully imported as well as the Interactions that have been created. .

## Setting up connections using the EDI Connection wizard

### About this task

Before you set up connections using the EDI Connection wizard, following must have been created:

- The internal partner
- At least one external partner
- An EDI Business ID for each partner. In this wizard, an EDI Business ID is defined as a Freeform Business Identifier having the form *qq-xxxxxxxxx*, where *qq* is the 2 digit EDI Interchange Qualifier, and *xxxxxxxxx* is the 9 digit EDI Interchange Identifier.
- Destinations and Default Destinations
- Envelope profiles

Several additional configuration steps may be needed before EDI flows can be successfully run. The following are examples:

- Configure XML Formats (if you are sending or receiving XML)
- Configure Receivers with ROD splitters (if you are receiving ROD)
- Configure additional Connection Attributes for AS or AS2 (if you are using the AS packaging)

To create Connections using the EDI Connection wizard, complete the following steps:

1. Start the WebSphere Partner Gateway Console.
2. Click **Wizards**.
3. Click **EDI Connection Wizard**.
4. Click the type of task to configure (**Send an EDI document to an EDI partner** or **Receive an EDI document from an EDI Partner**), and then click **Next**.
5. Depending on whether you selected **Receive an EDI document from an EDI partner** or **Send an EDI document to an EDI partner**, enter the source or target partner, and click **Search**.
6. Select a source or target partner from the drop-down list, and click **Next**.
7. Select the general properties for your source or target partner. If the Syntax is EDI, you must also specify EDI properties. When you have selected all the properties you want, click **Next**.

**Note:**

a. The TA1 and FA properties are visible only if source is an external partner. The FA required time is visible only if target is an external partner.

b. The EDI Connection Wizard contains a list of common values to be used as EDI delimiter values. If you want to use a value that is not in the list provided, you must edit the connection attribute by hand after completing the wizard. You can edit the connection attributes by clicking **Account Admin > Connections** .

c. You are forced to specify a Destination for each Operation Mode. This means that you cannot select the blank ("No Destination Selected") option. Forcing this additional Connection configuration does not negatively affect most document sending or receiving situations. If you need remove the Destination specification from the Connection, you can do so after completing the wizard by clicking **Account Admin > Connections**.

8. Select the source or target **Validation Map**, **Action**, and **Transformation Map** for the source or target partner. The map descriptions display after you select a map. Package is blank to prevent confusion in cases like EDI using the AS package. When you have selected these, click **Next**.
9. Review the suggested connections, click **Attributes**, **Actions**, or **Destinations** to review those settings.

   **Note:** Connections that already exist and are not being created are grayed out. These connections also, have an Exists icon beside them and do not have a Create check box. If connections already exist, they are not overwritten by this wizard. In this case, a warning appears explaining this situation.
   If the connections need to be modified, click **Back**. When you are satisfied by the connections listed, click **Finish**. If they need to be modified, click **Back**. and a final window displays showing the connections that you have successfully created.

## Defining document exchanges manually

The EIF Import wizard and the EDI Connection wizard can help you define document exchanges (for more information on these wizards see "Defining document exchanges using wizards" on page 191. However, you can also define documents manually. This section provides a high-level overview of the tasks you need to perform to establish the exchange of documents for EDI interchanges entering the hub, documents or transactions transformed at the hub, and for EDI interchanges being sent from the hub. The steps shown in the following sections are general and apply only to the importing of maps and setting up of interactions. The general steps for enabling B2B capabilities for partners (for all types of document exchanges) are described in "Setting up B2B capabilities" on page 26. The general steps for managing connections (for all types of document exchanges) are described in Chapter 12, "Managing connections," on page 231. If you want to see a comprehensive example of an EDI document exchange, from the importing of maps all the way through the management of connections, refer to Chapter 20, "EDI examples," on page 317. The appendix includes the following specific examples:

- " EDI to ROD example" on page 317
- " EDI to XML example" on page 330
- " ROD to EDI example" on page 342
- " XML to EDI example" on page 335

## Importing maps manually

### About this task

Transformation maps for EDI, XML, or record-oriented-data (ROD) documents can be created with the Data Interchange Services client program. The Data Interchange Services client is a program used to create and maintain XML schema document definitions, XML DTD document definitions, EDI standards, ROD document definitions, and maps.

WTX maps are created using the WTX Design studio and imported into WebSphere Partner Gateway.

The Data Interchange Services client is a separately installed program that is included on the WebSphere Partner Gateway media but that typically resides on another computer. The Data Interchange Services mapping specialist creates a map that specifies how the elements in one document are moved to the elements in another, different document. In addition to having instructions that explain how to convert a document from one format to another, Data Interchange Services must also know the layout, or format, of the source and target document. In Data Interchange Services the layout of a document is a *document definition*.

When the transformation map is imported into WebSphere Partner Gateway, the document definitions created in Data Interchange Services are displayed as document definitions (package, protocol, and document type) on the Transformation Map and Manage Document Definitions page.

For example, if you are converting an XML document to an X12 transaction, you import the map that defines the XML and X12 transaction document definitions and the transformation that is to take place.

There are two methods for receiving the map files from the Data Interchange Services. If the Data Interchange Services client has a direct connection to the WebSphere Partner Gateway database, the Data Interchange Services mapping specialist can export the file directly to the database. A more likely scenario is that you will receive the files in e-mail or as an FTP transfer. If the files are transferred to you through FTP, note that they must be in binary form.

If an error occurs during the export of a map from the Data Interchange Services client, you might still see the map name in the Community Console. The map cannot be used to translate documents. You will need to advise the Data Interchange Services client mapping specialist of the export problem and ask the mapping specialist to re-export the map before it can be used to translate documents.

To import a map, perform the following steps:

1. Open a command window.
2. Enter the following command or script:
   - On a UNIX system:

     `<ProductDir>/bin/bcgDISImport.sh <control_string_map>`
   - On a Windows system:

     `<ProductDir>\bin\bcgDISImport.bat <control_string_map>`

     where *<database_user_ID>* and *<password>* are the values that you used when you installed the database as part of the WebSphere Partner Gateway

installation. The <*control_string_map*> is the complete path of the map control string file exported from Data Interchange Services client.

3. For transformation maps, verify that the document definition was imported.
   a. Click **Hub Admin > Hub Configuration > Maps > Transformation Maps**.
   b. From the Transformation Maps page, click the **View details** icon next to the map from Data Interchange Services. You will notice that the document definitions for the source and target are displayed, indicating the format in which the document will be received at the hub and the format in which it will be delivered from the hub.
   c. Click **Hub Admin > Hub Configuration > Document Definition**.
   d. Expand the packages and protocols associated with the document definitions you saw on the Transformation Maps page to verify that the document types are displayed on the Manage Document Definitions page.

You can use validation maps in conjunction with transformation maps to add additional EDI Standards validation to any translation process involving EDI Standards. Validation maps give you complete control over the validation of an EDI document.

Note that transformation and validation maps exported from the Data Interchange Services client or imported with the bcgDISImport utility cannot be downloaded from the WebSphere Partner Gateway Community Console. The Data Interchange Services client mapping specialist administers these maps by connecting to the WebSphere Partner Gateway database through the Data Interchange Services client.

## Importing WTX maps

### About this task

The WTX maps created using WTX design studio must be imported into WebSphere Partner Gateway, so it can be associated to a particular participant connection. You need to manually create a DFD. The created DFDs are exported from WTX design studio in the form of a map that is complied for the native operating system. To import this into WebSphere Partner gateway, navigate to **hubadmin > Maps > transformation maps** and click **Create**. The imported map will be stored in common file system under a specific folder dedicated for WTX maps (common/maps).

## Importing WDI standard EIF

### About this task

In order to perform validation of EDI transactions in WebSphere Partner Gateway, the compiled form of the EDI standard must be available in WebSphere Partner Gateway. To create this compiled standard control string, perform the following:

1. Download the EDI standard from WDI support website.
2. Create a data-transformation map for transformation and select the EDI transaction that you want to validate in WebSphere Partner Gateway. For example, if you want to validate transaction 810 of X12V4R1, create a data transformation map from X12V4R1-810 to X12V4R1-810.
3. Map just one mandatory segment and compile the transformation map.
4. Export the Data transformation map control string into the document manager database. This will also export the compiled standard into the document manager database, which can be used for validation.

**Note:** Alternatively, there are some sample EIFs provided that includes just the compiled standard control string.

## Setting up an EDI to EDI flow

### About this task

This section describes interactions needed to receive an EDI interchange, de-envelope the interchange, transform a transaction from one EDI format to another, envelope the transaction, and deliver it.

1. Verify that a document definition exists for the EDI interchange that is received at the hub. Remember that after the interchange is de-enveloped, the original envelope does not continue to be processed. In other words, it has no delivery point. Therefore, you will use **N/A** for Package on the target interaction.
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Check to see whether a document definition already exists. For example, if a partner will be sending an EDI interchange in AS packaging, EDI-X12 protocol, and ISA document type, the definition is already available. Similarly, an N/A/EDI-X12/ISA document definition already exists.
   c. Enter a value (or select the value from the list) for any attribute you want associated with the profile. For example, if you want to specify that the envelope should be discarded if errors are found with any of the transactions, click the **Edit attribute values** icon next to **Document Definitions**. In the **Discard Envelope if Any Errors** row, select **Yes** from the list.
   d. If a document definition does not exist, create one by selecting the Package, Protocol, and Document Type.

   **Note:** You cannot use the attribute Discard Envelope on Error when the action in the connection is EDI Interchange Validate.
2. Create an interaction for the interchange.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Select the source and target document definitions. Except for the packaging (which will be **N/A** for the target), the document definitions will be the same.
   d. Select **EDI De-envelope** from the Action list.
3. Import the transformation map that provides document definitions of the EDI transactions and that describes how the transaction is transformed from one EDI format to another. See " Importing maps manually" on page 194.

   If the interchange contains more than one transaction, repeat this step for each transaction.
4. If you want to edit attributes of the document definitions associated with the map, perform the following steps:
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Click the **Edit attribute values** icon next to the protocol. For EDI protocols, you see a long list of attributes that you can set.
   c. Enter a value (or select the value from the list) for any attribute you want associated with the protocol.
   d. Click the **Edit attribute values** icon next to the document definition. You generally see a smaller list of attributes than those associated with the protocol.

e. Enter a value (or select the value from the list) for any attribute you want associated with the document type. For example, you can change the **Validation Map** associated with the document type.

   Make sure you select an envelope profile for the transaction.

5. Create an interaction for the map you just imported.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Under **Source**, select the document type associated with the transaction. Expand the package and protocol and select the document type. This will typically be **N/A** (because the transaction itself did not originate from a partner), the protocol defined in the map (for example, **X12V4R1**) and the actual EDI document defined in the map (for example, **850**).
   d. Under **Target**, select the document definition for the transformed document. Expand the package and protocol and select the document type. Because the transaction will be enveloped (and will, therefore, not be directly delivered to a partner), the packaging will again be **N/A**.
   e. From the transformation map list, select the map that defines how to transform this document.
   f. From the Action list, select **EDI Validate and EDI Translate** for native WDI. In case of WTX, select **EDI Validate and WTX Transformation**.
6. Verify that a document definition exists for the EDI interchange that is being sent from the hub, and set any attributes that you want associated with the interchange.
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Check to see whether a document definition already exists. The source package will be N/A, with the protocol and document type matching the protocol and document type used to deliver the interchange. For example, if the interchange will be delivered as AS/EDI-X12/ISA, the source will be N/A/EDI-X12/ISA.
   c. Edit any attributes that apply to the interchange that is being delivered.
   d. If a document definition does not exist, create one by selecting the Package, Protocol, and Document Type.
7. Create an interaction for the EDI interchange that is sent from the hub after the transaction is transformed.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Select the source and target documents. Except for the packaging (which will be **N/A** for the source document), the document definitions will be the same.
   d. Select **Pass Through** from the **Action** list.

To add an acknowledgment to the flow, see “Setting up acknowledgments” on page 203.

After setting up the interactions, create B2B capabilities for the partners.

- For the source partner, enable three document definitions (under **Set Source**)--one for the source document type, one for the EDI transaction, and one for the envelope.

- For the target partner, enable three document definitions (under **Set Target**)--one for the de-enveloped document type, one for the transformed EDI transaction, and one for the EDI envelope.

The detailed steps for creating B2B capabilities are described in "Setting up B2B capabilities" on page 26.

After setting up B2B capabilities for the partners, create connections. You need three connections:

- One for the envelope from the source partner to the hub.
- One for the source EDI transaction to the target EDI transaction.
- One for the envelope from the hub to the target partner.

The detailed steps for creating connections are described in Chapter 12, "Managing connections," on page 231.

## Setting up an EDI to XML or ROD flow

### About this task

This section describes interactions needed to receive an EDI interchange, de-envelope the interchange, transform a transaction from an EDI format to an XML or ROD document, and deliver it.

**Note:** For a comprehensive example of the EDI to XML flow, see " EDI to XML example" on page 330. For a comprehensive example of the EDI to ROD flow, see " EDI to ROD example" on page 317.

1. Verify that a document definition exists for the EDI interchange that is received at the hub. Remember that after the interchange is de-enveloped, the envelope does not continue to be processed. In other words, it has no delivery point. Therefore, you will use **N/A** for Package on the target interaction.
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Check to see whether a document definition already exists. For example, if a partner will be sending an EDI interchange in AS packaging, EDI-X12 protocol, and ISA document type, the definition is already available. Similarly, an N/A/EDI-X12/ISA document definition already exists.
   c. If a document definition does not exist, create one.
2. Create an interaction for the EDI interchange that is received at the hub.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Select the source and target documents. Except for the packaging (which will be **N/A** for the target), the document definitions will be the same.
   d. Select **EDI De-envelope** from the Action list.
3. Import the transformation map that provides document definitions of the EDI transaction and the XML or ROD document and describes how the transaction is transformed into the XML or ROD document. See " Importing maps manually" on page 194.

   If the interchange contains more than one transaction, repeat this step for each transaction.
4. Create an interaction for the map you just imported.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.

c. Under **Source**, select the document type associated with the transaction. Expand the package and protocol and select the document type. This will typically be **N/A** (because the transaction itself did not originate from a partner), the protocol defined in the map (for example, **X12V4R1**) and the actual EDI document defined in the map (for example, **850**).

d. Under **Target**, select the document definition for the transformed (XML or ROD) document. Expand the package and protocol and select the document type.

e. From the transformation map list, select the map that defines how to transform this document.

f. From the Action list, select **EDI Validate and EDI Translate** if it is native WDI. In case of WTX, select **EDI Validate and WTX Transformation**.

To add an acknowledgment to the flow, see "Setting up acknowledgments" on page 203.

After setting up the interactions, create B2B capabilities for the partners.

- For the source partner, enable two document definitions (under **Set Source**)--one for the envelope and one for the EDI transaction.
- For the target partner, enable two document definitions (under **Set Target**)--one for the EDI envelope and one for the XML or ROD document.

The detailed steps for creating B2B capabilities are described in "Setting up B2B capabilities" on page 26.

After setting up B2B capabilities for the partners, create connections. You need two connections:

- One for the envelope from the source partner to the hub.
- One for the source EDI transaction to the XML or ROD document.

The detailed steps for creating connections are described in Chapter 12, "Managing connections," on page 231.

## Setting up an XML or ROD to EDI flow

### About this task

This section describes interactions needed to receive an XML or ROD document, transform it into an EDI transaction, envelope the transaction, and deliver it.

**Note:** For a comprehensive example of the XML to EDI flow, see " XML to EDI example" on page 335. For a comprehensive example of the ROD to EDI flow, see " ROD to EDI example" on page 342.

1. Import the transformation map that provides document definitions of the XML or ROD document and EDI transaction and describes how the document is transformed to the EDI transaction. See " Importing maps manually" on page 194.
2. Create an interaction for the map you just imported.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Under **Source**, select the document definition associated with the XML or ROD document. Expand the package and protocol and select the document type.

d. Under **Target**, select the document type associated with the EDI transaction. Expand the package and protocol and select the document type. Because the transaction will not be delivered directly (it will be put into an envelope before delivery), **N/A** will be listed for Package.

e. From the transformation map list, select the map that defines how to transform this document.

f. From the Action list, select **XML Translate and EDI Validate** or **ROD Translate and EDI Validate** for native WDI. In case of WTX, select **WTX Transformation**.

3. Verify that a document definition exists for the EDI interchange that is being sent from the hub, and set any attributes that you want associated with the interchange.
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Check to see whether a document definition already exists. **N/A** should be used for Package for the source document (the interchange being sent from the hub).
   c. Edit any attributes that apply to the interchange that is being delivered.
   d. If a document definition does not exist, create one by selecting the Package, Protocol, and Document Type.
4. Create an interaction for the EDI interchange that is sent from the hub after the document is transformed.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Select the source and target documents. The source and target documents have different packaging (the source document has a package of N/A), but the protocol (for example, EDI-X12) and the document type (for example, ISA) should be the same.
   d. Select **Pass Through** from the Action list.

After setting up the interactions, create B2B capabilities for the partners.

- For the source partner, the number of document definitions you need to set (under **Set Source**) varies, depending on the document type.
  - For example, for an XML document in which the document type is ICGPO and the translated EDI transaction is MX12V3R1, enable three document definitions (under **Set Source**)--one for the XML (ICGPO) document, one for the EDI transaction (MX12V3R1), and one for the envelope being sent from the hub.
  - For other XML documents and for ROD documents, enable two document definitions (under **Set Source**)--one for the XML or ROD document and one for the envelope being sent from the hub.
- For the target partner, enable two document definitions (under **Set Target**)--one for the EDI transaction and one for the EDI envelope that is received. For the EDI transaction, click the **Edit attribute values** icon next to the protocol, and specify an envelope profile. You can specify other attributes as well.

The detailed steps for creating B2B capabilities are described in "Setting up B2B capabilities" on page 26.

After setting up B2B capabilities for the partners, create connections. You need two connections:

- One for the source XML or ROD document to EDI transaction.

- One for the envelope from the hub to the partner.

The detailed steps for creating connections are described in Chapter 12, "Managing connections," on page 231.

## Setting up multiple XML or ROD documents in one file to EDI flow

### About this task

This section describes interactions needed to receive multiple XML or ROD documents in one file, transform the documents into EDI transactions, envelope the transactions, and deliver the EDI interchange.

1. Import the transformation map that provides the document definitions of the XML or ROD documents and the EDI transactions and that describes the transformation. See " Importing maps manually" on page 194.
2. Create an interaction for the source and target documents.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. For native WDI, select the source and target documents, and select **XML Translate and EDI Validate** or **ROD Translate and EDI Validate** from the Action list. For WTX, select **WTX Transformation** and **EDI validate**.
3. Repeat step 2 for the source document and each target document produced by the transformation map.
4. Verify that a document definition exists for the EDI interchange that is being sent from the hub, and set any attributes that you want associated with the interchange.
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Check to see whether a document definition already exists. The source will be N/A, with the protocol and document type matching the protocol and document type used to deliver the interchange. For example, if the interchange will be delivered as AS/EDI-X12/ISA, the source will be N/A/EDI-X12/ISA.
   c. Edit any attributes that apply to the interchange that is being delivered.
   d. If a document definition does not exist, create one by selecting the Package, Protocol, and Document Type.
5. Create an interaction for the EDI interchange that is sent from the hub after the transaction is transformed.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Select the source and target documents. The source and target documents have different packaging (the source document has a package of N/A), but the protocol (for example, EDI-X12) and the document type (for example, ISA) should be the same.
   d. Select **Pass Through** from the Action list.

After setting up the interactions, create B2B capabilities for the partners.

- For the source partner, the number of document definitions you need to set (under **Set Source**) varies, depending on the document type.
  – For example, for an XML document in which the document type is ICGPO and the translated EDI transaction is MX12V3R1, enable three document

definitions (under **Set Source**)--one for the XML (ICGPO) document, one for the EDI transaction (MX12V3R1), and one for the envelope being sent from the hub.

– For other XML documents and for ROD documents, enable two document definitions (under **Set Source**)--one for the XML or ROD document and one for the envelope being sent from the hub.

The detailed steps for creating B2B capabilities are described in "Setting up B2B capabilities" on page 26.

After setting up B2B capabilities for the partners, create connections. You need several connections:

- One for each XML or ROD document that is transformed into an EDI transaction.
- One for the envelope from the hub to the partner.

The detailed steps for creating connections are described in Chapter 12, "Managing connections," on page 231.

## Setting up an XML to ROD or ROD to XML document flow

### About this task

This section describes interactions needed to receive an XML or ROD document, transform it into the other document type (XML to ROD or ROD to XML) and deliver it.

1. Import the transformation map that provides document definitions of the XML and ROD documents and that describes how the documents are transformed. See " Importing maps manually" on page 194.
2. Click **Hub Admin > Hub Configuration > Maps > Transformation Maps** and click the **View details** icon next to the map you just imported.
3. Create an interaction for the map you just imported.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
4. Select the source and target documents, and select **WTX transformation** for WTX or **ROD Translate and EDI Validate** from the Action list.

After setting up the interactions, create B2B capabilities for the partners.

- For the source partner, enable document definitions (under **Set Source**) for the XML or ROD document.
- For the target partner, enable document definitions (under **Set Target**) for the XML or ROD document.

The detailed steps for creating B2B capabilities are described in "Setting up B2B capabilities" on page 26.

After setting up B2B capabilities for the partners, create connections. You need one connection--for the XML to ROD flow or for the ROD to XML flow. The detailed steps for creating connections are described in Chapter 12, "Managing connections," on page 231.

## Setting up an XML to XML or ROD to ROD flow

### About this task

This section describes interactions needed to receive an XML or ROD document, transform it into a document of the same type (XML to XML or ROD to ROD) and deliver it.

1. Import the transformation map that provides document definitions of the XML or ROD documents and that describes how the documents are transformed. See " Importing maps manually" on page 194.
2. Click **Hub Admin > Hub Configuration > Maps > Transformation Maps** and click the **View details** icon next to the map you just imported.
3. Create an interaction for the map you just imported.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
   b. In the **Manage Interactions** screen, click **Create Interaction link**.
   c. Select the source and target documents.
   d. For native WDI, select **XML Translate and EDI Validate** or **ROD Translate and EDI Validate** from the Action list. For WTX, select **WTX Transformation** and **EDI Interchange Validation**.

After setting up the interactions, create B2B capabilities for the partners.

- For the source partner, enable a document definition (under **Set Source**) for the XML or ROD document.
- For the target partner, enable a document definition (under **Set Target**) for the XML or ROD document.

The detailed steps for creating B2B capabilities are described in "Setting up B2B capabilities" on page 26.

After setting up B2B capabilities for the partners, create connections. You need one connection--for the XML to XML flow or for the ROD to ROD flow. The detailed steps for creating connections are described in Chapter 12, "Managing connections," on page 231.

## Setting up acknowledgments

This section describes how to set up interactions to send acknowledgments of interchange or transaction receipt to the originator of the document.

### Functional acknowledgments

Functional acknowledgment maps are used to provide generation of functional acknowledgments when responding to EDI documents received from a partner. WebSphere Partner Gateway provides a set of functional acknowledgment maps that produce the commonly used EDI functional acknowledgments. The mapping specialist can also create FA and validation maps, in which case these maps would be uploaded to WebSphere Partner Gateway.

**Note:** A functional acknowledgment map should be created only when a custom functional acknowledgment is required.

In addition to the functional acknowledgment maps provided with WebSphere Partner Gateway, the &FUNC_ACK_METADATA_DICTIONARY protocol and associated &FUNC_ACK_META are provided. They are listed under **Package: None** in the Document Definitions page. &FUNC_ACK_META is the source

document definition for all functional acknowledgment maps. This map provides the structure of the functional acknowledgment. A functional acknowledgment flows to partners, and the functional acknowledgment map tells the system how the acknowledgment should be generated. The name of the source document definition cannot be changed. The Data Interchange Services client mapping specialist cannot create a functional acknowledgment map without this document definition in your database.

The target document definition in a functional acknowledgment map describes the layout of the functional acknowledgment. It must be an EDI document definition with a name of 997, 999, or CONTRL.

The following functional acknowledgment maps are installed with WebSphere Partner Gateway and appear on the Manage Document Definitions page under **Package: N/A**:

*Table 29. Product-provided functional acknowledgment maps*

| Protocol | Document Type | Description |
|---|---|---|
| &DTCTL21 | CONTRL | Functional Acknowledgement CONTRL – UN/EDIFACT Version 2 Release 1 (D94B) |
| &DTCTL | CONTRL | Functional Acknowledgement CONTRL – UN/EDIFACT prior to D94B |
| &DT99933 | 999 | Functional Acknowledgement 999 – UCS Version 3 Release 3 |
| &DT99737 | 997 | Functional Acknowledgement 997 – X12 Version 3 Release 7 |
| &DT99735 | 997 | Functional Acknowledgement 997 – X12 Version 3 Release 5 |
| &DT99724 | 997 | Functional Acknowledgement 997 – X12 Version 2 Release 4 |

In addition, the &X44TA1 protocol (with an associated TA1 document type) are listed under **Package: N/A**. This map is used to generate a TA1. TA1 is a functional acknowledgment that is generated for incoming X12 interchanges.

The &WDIEVAL protocol (with an associated X12ENV) is also provided under **Package: N/A**.

Like EDI transactions, functional acknowledgments are always put into an EDI interchange before being delivered.

### TA1 acknowledgments

TA1 is an EDI segment that provides X12 interchange acknowledgment. It acknowledges the receipt and syntactical correctness of an X12 interchange header and trailer (ISA and IEA) pair. The sender can request a TA1 from the receiver by setting element 14 of the ISA Interchange Control Header to **1**. The interchange control number of a TA1 is matched to a previously transmitted X12 interchange with the same control number to complete the acknowledgment process.

Like EDI transactions and functional acknowledgments, TA1s are always put into an EDI interchange before being delivered.

## Adding an acknowledgment to the document type

### About this task

To add an acknowledgment to a flow, perform the following steps:

### Procedure

1. If the functional acknowledgment map is not supplied by WebSphere Partner Gateway, import the map from the Data Interchange Services client. See " Importing maps manually" on page 194.
2. Associate the FA map with a document definition:
   a. Click **Hub Admin > Hub Configuration > Maps > EDI FA Maps**.
   b. Click the **View details** icon next to the map.
   c. Click the **Expand** icon next to a package to individually expand to the appropriate level (for example, expand the **Package** and **Protocol** folders and select the transaction).
   d. Click **Save**.
3. Create an interaction for the map you just imported.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions** link.
   b. In the **Manage Interactions** screen, click **Create Interaction** link.
   c. Under **Source**, select the document type associated with the functional acknowledgment. Expand the package and protocol and select the document type.
   d. Under **Target**, select the same values.
   e. From the Action list, select **Pass Through**.
4. Verify that a document definition exists for the EDI interchange that is being sent from the hub, and set any attributes that you want associated with the interchange.
   a. Click **Hub Admin > Hub Configuration > Document Definition**.
   b. Check to see whether a document definition already exists. The source will be N/A, with the protocol and document type matching the protocol and document type used to deliver the interchange. For example, if the interchange will be delivered as AS/EDI-X12/ISA, the source will be N/A/EDI-X12/ISA.
   c. Edit any attributes that apply to the interchange that is being delivered.
   d. If a document definition does not exist, create one by selecting the Package, Protocol, and Document Type.
5. Create an interaction for the EDI interchange that is sent from the hub after the document is transformed.
   a. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions** link.
   b. In the **Manage Interactions** screen, click **Create Interaction** link.
   c. Select the source and target documents.
   d. Select **Pass Through** from the **Action** list.

### Results

After setting up the interactions, create B2B capabilities for the partners. Note that the target partner in a functional acknowledgment transmission is the source partner of the original EDI document.

- For the source partner, enable document definitions (under **Set Source**) for the functional acknowledgment. Also enable a document definition for the envelope that is being sent from the hub.
- For the target partner, enable a document definition (under **Set Target**) for the functional acknowledgment. Also enable a document definition for the EDI envelope that is received.

  For the functional acknowledgment, click the **Edit attribute values** icon next to the protocol, and specify an envelope profile.

The detailed steps for creating B2B capabilities are described in "Setting up B2B capabilities" on page 26.

After setting up B2B capabilities for the partners, create connections. You need two connections:

- One for the functional acknowledgment.
- One for the envelope from the hub to the partner.

The detailed steps for creating connections are described in Chapter 12, "Managing connections," on page 231.

**Related concepts**

Chapter 12, "Managing connections," on page 231

**Related tasks**

" Importing maps manually" on page 194

"Setting up B2B capabilities" on page 26

## Viewing EDI interchanges and transactions

### About this task

As mentioned earlier in this chapter, you use the Document Viewer to display information about the EDI interchanges and transactions that make up a document flow. You can display raw documents and associated document processing details and events using specific search criteria. This information is useful if you are trying to determine whether an EDI interchange was successfully delivered or to determine the cause of a problem.

To display the Document Viewer complete the following:

1. Click **Viewers > Document Viewer**.
2. Select the appropriate search criteria.
3. Click **Search**.

See the *WebSphere Partner Gateway Administrator Guide* for information on using the Document Viewer.

## OpenPGP limitations while receiving and sending EDI documents over different transport protocols

While receiving EDI documents, the business IDs are determined from the content, and they have to match the business IDs as determined from the packaging or the folder structure. The following are the limitations of receiving EDI data over different transport protocols:

1. While receiving a document over HTTP, basic authentication determines the sending partner. If 'To' parameter is used, then it determines the business ID of the receiving partner. The 'X-receiver' transport header can also be used to

identify the recipient partner. It has to contain the business id of the recipient partner. If the recipient partner is not specified, then the default internal partner is considered as the recipient. Basic authentication contains the user id and password. It is recommended to use HTTP(S) with server authentication and Basic authentication.

2. When receiving a document over FTP(S), the sending partner is determined by the WebSphere Partner Gateway specific folder structure, configured for FTP(S) receivers.
3. In case of Binary documents, when the document is received over SFTP, the sending partner is determined based on the configuration values, provided in the attached Generic pre-process handler of SFTP receiver.

# Chapter 11. Creating destinations

After you create the partners, you define destinations for the partners. Destinations define entry points into the partner's system.

This chapter covers the following topics:



**Note:** You should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Overview of destinations

WebSphere Partner Gateway uses destinations to route documents to their proper destination. The recipient can be an external partner or the internal partner. The outbound transport protocol determines which information is used during

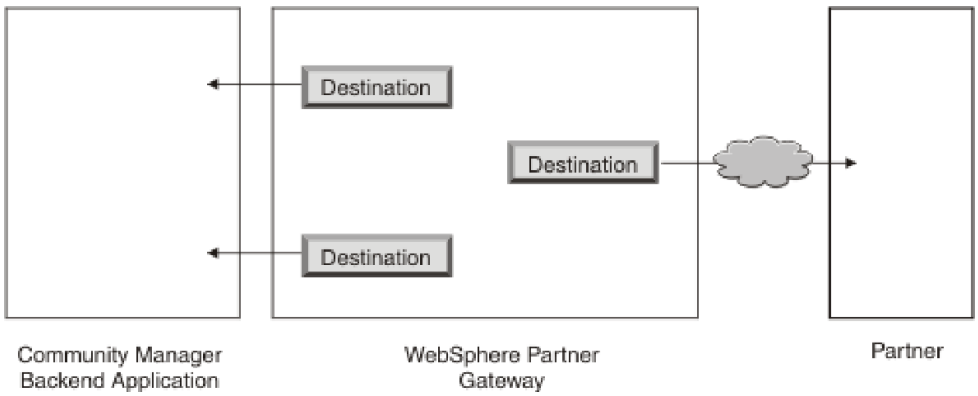



*Figure 34. Destinations to internal partner and external partners*

destination configuration.

The following transports are supported (by default) for partner destinations:

- HTTP/1.1
- HTTPS/1.0
- HTTPS/1.1
- FTP
- FTPS
- JMS
- SMTP

  **Note:** You can define an SMTP destination for external partners only (not for the internal partner).
- SFTP
- File directory
- FTP Scripting

You can also specify a user-defined transport, which you upload during the creation of the destination.

As the hub administrator, you can set up the destinations for your partners, or the partners can perform this task themselves. In this chapter, you will see how to perform the task for the partners. For managing destinations, see *Hub Administration tasks Chapter of Administrator Guide*.

## Setting up global transport values

### About this task

You set global transport attributes that apply to all FTP Scripting destinations. If you are not defining any FTP Scripting destinations, this section does not apply to you.

The FTP Scripting transport uses a locking mechanism that prevents more than one FTP Scripting instance from accessing the same destination at the same time. Default values are supplied for such things as how long a gateway instance waits to obtain the lock and how many times it attempts to retrieve it if the lock is in use. You can use these default values or change them.

1. Click **Account Admin > Profiles**.
2. Click **Destinations**.
3. Select **Global Transport Attributes** from the Destination Details.

   If you updated either **Maximum Lock Time (Seconds)** or **Maximum Queue Time (Seconds)** when you specified global transport values during the creation of targets, those updated values are reflected here.
4. If the default values are appropriate for your configuration, click **Cancel**. Otherwise, continue with the remaining steps in this section.
5. Click the **Edit** icon next to **FTP Scripting Transport**.
6. To change one or more of the values, type the new value or values. You can change:
   - **Lock Retry Count**, which indicates how many times the destination will attempt to obtain a lock if the lock is currently in use. The default is 3.
   - **Lock Retry Interval (Seconds)**, which indicates the amount of time that will elapse between attempts to obtain the lock. The default is 260 seconds.

- **Maximum Lock Time (Seconds)**, which indicates how long the destination can hold the lock. The default is 240 seconds (unless you changed it when creating targets).
- **Maximum Queue Age (Seconds)**, which indicates how long the target will wait in a queue to obtain the lock. The default is 740 seconds (unless you changed it when creating targets).

7. Click **Save**

## Configuring a forward proxy

### About this task

For the HTTP transport, you can set up forward proxy support so that documents are sent through a configured proxy server. With WebSphere Partner Gateway, you can set up the following types of support:

- Proxy support over HTTP
- Proxy support over HTTP with authentication
- Proxy support over SOCKS

**Note:** WebSphere Partner Gateway connects to Proxy server only on HTTP port.

After you set up a forward proxy, you can make it global for the transport by making it the default destination (for example, all HTTP destinations make use of the forward proxy).

To set up a forward proxy, perform the following steps:

1. Click **Account Admin > Profiles > Partner**.
2. Click **Destinations**.
3. Click **Forward Proxy Support**.
4. On the Forward Proxy List page, click **Create**.
5. Type a name for the proxy.
6. Optionally, type a description of the proxy.
7. Select the transport type from the list.

   **Note:** The available transports are HTTP and HTTPS.
8. Type the following information. Enter either Proxy Host and Proxy Port *or* Socks Proxy Host and Socks Proxy Port.
   - For **Proxy Host**, type the proxy server to use (for example: http://proxy.abc.com).
   - For **Proxy Port**, type the port number.
   - If the proxy server requires a user name and password, specify them in the **User Name** and **Password** fields.
   - For **Socks Proxy Host**, type the SOCKS proxy server to use.
   - For **Socks Proxy Port**, type the port number.
9. Select the check box if you want this proxy to be the default proxy (which can be used by any partner that has proxy support specified).
10. Click **Save**.

**Note:** The HTTP tunneling technique is used in forward proxy, but there is no support for Secure Forward Proxy. HTTP Tunnel is created with the Proxy Server. You need to check the connectivity before passing any type of data (HTTP or

HTTPS) to the end partner. The data is SSL encrypted. The port used for Forward Proxy must be HTTP port 80. It is basically a passthru of the SSL handshake between WebSphere Partner Gateway and the Partner.

# Setting up an HTTP destination

## About this task

You set up an HTTP destination so that documents can be sent from the hub to your partner's IP address. When you set up an HTTP destination, you can also specify that documents be sent through a configured proxy server.

To begin the process of creating an HTTP destination, use the following procedure.

1. Click **Account Admin > Profiles**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

# Destination Details

## About this task

From the **Destination List** page, perform the following steps:

1. Type a name to identify the destination. This is a required field. This is the name that will appear on the list of destinations.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.

# Destination configuration

## About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. Optionally, select a proxy server to be used. The **Forward Proxy List** includes any proxy servers that you have created, including the default proxy server. The default value for this field is **Use default forward proxy**. If you want the selected partner to use a different proxy server, select that server from the list. If you do not want to use this feature with the selected partner, select **Use no forward proxy**.
2. In the **Address** field, enter the URI where the document will be delivered. This field is required.

   The format is: http://*<server_name>*:*<optional_port>*/*<path>*

   An example of this format is:

   ```
   http://anotherserver.ibm.com:57080/bcgreceiver/Receiver
   ```

   **Note:** If you are specifying an IPv6 address, provide the numeric format, not the machine name or host name.

Examples of IPv6 addresses include:

```
http://[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]:80/index.html
http://[1080:0:0:0:8:800:200C:417A]/index.html
http://[3ffe:2a00:100:7031::1]
http://[1080::8:800:200C:417A]/foo
http://[::192.9.5.5]/ipng
http://[::FFFF:129.144.52.38]:80/index.html
http://[2010:836B:4179::836B:4179]
```

When you are setting up a destination to be used for a Web service, specify the private URL supplied by the Web service provider. This is where WebSphere Partner Gateway will invoke the Web service when it acts as a proxy for the Web service provider.

3. Optionally enter a user name and password, if a user name and password are required to access the HTTP server.
4. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
5. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.
6. In the **Number of Threads** field, enter the number of documents that can be processed simultaneously. The default is 3.
7. In the **Validate Client IP** field, select **Yes** if you want the IP address of the sender to be validated before the document is processed. Select **No** otherwise. The default is **No**.
8. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure is about to occur because the number of retries has been exhausted. Select **No** otherwise. The default is **No**.

   When you select **Auto Queue**, all documents remain queued until the destination is placed online manually.
9. In the **Connection Timeout** field, enter the number of seconds a socket will remain open with no traffic. The default is 120 seconds.
10. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers" on page 228. Otherwise, click **Save**.

## Setting up an HTTPS destination

### About this task

You set up an HTTPS destination so that documents can be sent from the hub to your partner's IP address. When you set up an HTTPS destination, you can also specify that documents be sent through a configured proxy server.

To create HTTPS destinations, use the following procedure.

1. Click **Account Admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

## Destination details

### About this task

From the Destination Details page, perform the following steps:

1. Type a name to identify the destination. This is a required field.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.
5. Select **HTTPS/1.0** or **HTTPS/1.1** from the **Transport** list.

## Destination configuration

### About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. Optionally, select a proxy server to be used. The **Forward Proxy List** includes any proxy servers that you have created, including the default proxy server. The default value for this field is **Use default forward proxy**. If you want the selected partner to use a different proxy server, select that server from the list. If you do not want to use this feature with the selected partner, select **Use no forward proxy**.
2. In the **Address** field, enter the URI where the document will be delivered. This field is required.

   The format is: https://*<server_name>:<optional_port>/<path>*

   For example:

   ```
   https://anotherserver.ibm.com:57443/bcgreceiver/Receiver
   ```

   **Note:** If you are specifying an IPv6 address, provide the numeric format, not the machine name or host name.

   Examples of IPv6 addresses include:

   ```
   https://[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]:80/index.html
   https://[1080:0:0:0:8:800:200C:417A]/index.html
   https://[3ffe:2a00:100:7031::1]
   https://[1080::8:800:200C:417A]/foo
   https://[::192.9.5.5]/ipng
   https://[::FFFF:129.144.52.38]:80/index.html
   https://[2010:836B:4179::836B:4179]
   ```
3. Optionally enter a user name and password, if a user name and password are required to access the secure HTTP server.
4. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
5. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.
6. In the **Number of Threads** field, enter the number of documents that can be processed simultaneously. The default is 3.
7. In the **Validate Client IP** field, select **Yes** if you want the IP address of the sender to be validated before the document is processed. Select **No** otherwise. The default is **No**.

8. In the **Validate Client SSL Cert** field, select **Yes** if you want the digital certificate of the sending partner to be validated against the business ID associated with the document. The default is **No**.
9. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure is about to occur because the number of retries has been exhausted. Select **No** otherwise. The default is **No**.

   When you select **Auto Queue**, all documents remain queued until the destination is placed online manually.
10. In the **Connection Timeout** field, enter the number of seconds a socket will remain open with no traffic. The default is 120 seconds.
11. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers" on page 228. Otherwise, click **Save**.

# Setting up an FTP destination

## About this task

To create an FTP destination, use the following procedure.

1. Click **Account Admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

**Note:** FTP passive mode is not supported. See "Setting up an FTP Scripting destination" on page 223 for passive support.

# Destination Details

## About this task

From the Destination Details page, perform the following steps:

1. Type a name to identify the destination. This is a required field.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.

# Destination configuration

## About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. In the **Address** field, enter the URI where the document will be delivered. This field is required.

   The format is: ftp://<*ftp_server_name*>:<*portno*>

   For example:

   `ftp://ftpserver1.ibm.com:2115`

   If you do not enter a port number, the standard FTP port is used.

**Note:** If you are specifying an IPv6 address, provide the numeric format, not the machine name or host name.

Examples of IPv6 addresses include:

```
ftp://[FEDC:BA98:7654:3210:FEDC:BA98:7654:3210]:21
ftp://[1080:0:0:0:8:800:200C:417A]:21
ftp://[3ffe:2a00:100:7031::1]:21
ftp://[1080::8:800:200C:417A]:21
ftp://[::192.9.5.5]:21
ftp://[::FFFF:129.144.52.38]:21
ftp://[2010:836B:4179::836B:4179]:21
```

2. Optionally enter a user name and password, if a user name and password are required to access the FTP server.
3. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
4. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.
5. In the **Number of Threads** field, enter the number of documents that can be processed simultaneously. The default is 3.
6. In the **Validate Client IP** field, select **Yes** if you want the IP address of the sender to be validated before the document is processed. Select **No** otherwise. The default is **No**.
7. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure is about to occur because the number of retries has been exhausted. Select **No** otherwise. The default is **No**.

   When you select **Auto Queue**, all documents remain queued until the destination is placed online manually.
8. In the **Connection Timeout** field, enter the number of seconds a socket will remain open with no traffic. The default is 120 seconds.
9. If you want the document to have its original name when it is sent to its destination, do not select **Use Unique File Name**. Otherwise, select this if you want WebSphere Partner Gateway to assign a name to the file.
10. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers" on page 228. Otherwise, click **Save**.

## Setting up an SMTP destination

### About this task

To create an SMTP destination, use the following procedure.

1. Click **Account Admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

## Destination Details

### About this task

From the Destination List page, perform the following steps:

1. Type a name to identify the destination. This is a required field.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.

## Destination configuration

### About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. In the **Address** field, enter the URI where the document will be delivered. This field is required.

   The format is: mailto:<*user@server_name*>

   For example:

   `mailto:admin@anotherserver.ibm.com`
2. Optionally enter a user name and password, if a user name and password are required to access the SMTP server.
3. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
4. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.
5. In the **Number of Threads** field, enter the number of documents that can be processed simultaneously. The default is 3.
6. In the **Validate Client IP** field, select **Yes** if you want the IP address of the sender to be validated before the document is processed. Select **No** otherwise. The default is **No**.
7. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure is about to occur because the number of retries has been exhausted. Select **No** otherwise. The default is **No**.

   When you select **Auto Queue**, all documents remain queued until the destination is placed online manually.
8. In the **Authentication Required** field, indicate whether a user name and password are supplied with the document. The default is **No**.
9. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers" on page 228. Otherwise, click **Save**.

---

# Setting up a JMS destination

### About this task

To create JMS destinations, use the following procedure.

1. Click **Account Admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.

3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

**Note:** For information on configuring the runtime libraries so that the requisite WebSphere MQ jar files are visible to WebSphere Partner Gateway, see "Configuring the runtime libraries" on page 40.

## Destination Details

### About this task

From the Destination List page, perform the following steps:

1. Type a name to identify the destination. This is a required field.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.

## Destination configuration

### About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. In the **Address** field, enter the URL where the document will be delivered. This field is required.

   For WebSphere MQ JMS, the format of the target URL is as follows:

   `file:///<user_defined_MQ_JNDI_bindings_path>`

   For example:

   ```
   file:///opt/JNDI-Directory in case of UNIX and
   file://c:/temp/ in case of Windows.
   ```

   The directory contains the ".bindings" file for the file-based JNDI. This file indicates to WebSphere Partner Gateway how to route the document to its intended destination.

   - For an internal JMS destination (that is, the destination to your back-end system), this should match the value you entered (the file system path to the bindings file) when you configured WebSphere Partner Gateway for JMS (step 5 on page 38). You can also specify the subfolder for the JMS context as part of the JMS provider URL.

     For example, without the JMS context, you would enter `c:/temp/JMS`. With the JMS context, you would enter `c:/temp/JMS/JMS`.
   - For partner destinations, the partner will probably provide the ".bindings" file.

   This field is required.
2. Optionally enter a user name and password, if a user name and password are required to access the JMS queue.
3. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
4. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.

5. In the **Number of Threads** field, enter the number of documents that can be processed simultaneously. The default is 3.
6. In the **Validate Client IP** field, select **Yes** if you want the IP address of the sender to be validated before the document is processed. Select **No** otherwise. The default is **No**.
7. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure is about to occur because the number of retries has been exhausted. Select **No** otherwise. The default is **No**.

   When you select **Auto Queue**, all documents remain queued until the destination is placed online manually.
8. In the **Authentication Required** field, indicate whether a user name and password are supplied with the document. The default is **No**.
9. In the **JMS Factory Name** field, enter the name of the Java class the JMS provider uses to connect to the JMS queue. This field is required.

   For internal JMS destinations, this name should match the one you specified with the `define qcf` command when you created the bindings file (step 4 on page 39).

   If you entered the subfolder for the JMS context in step 1 on page 218, enter only the factory name here (for example, `Hub`). If you did not enter the subfolder for the JMS context in the **Address** field, specify the subfolder before the factory name (for example, `JMS/Hub`).
10. In the **JMS Message Class** field, enter the message class. The choices are any valid JMS Message class, such as TextMessage or BytesMessage. This field is required.
11. In the **JMS Message Type** field, enter the type of message. As the Receiver component decides the JMS message type mapping, the value of JMS Message type is optional.
12. In the **Provider URL Packages** field, enter the name of the classes (or JAR file) that Java uses to understand the JMS context URL. This field is optional. If you do not specify a value, the file system path to the bindings file is used.
13. In the **JMS Queue Name** field, enter the name of the JMS queue where documents are to be sent. This field is required.

    For internal JMS destinations, this name should match the one you specified with the `define q` command when you created the bindings file (step 4 on page 39).

    If you entered the subfolder for the JMS context in step 1 on page 218, enter only the queue name here (for example, `outQ`). If you did not enter the subfolder for the JMS context in the JMS provider URL, specify the subfolder before the factory name (for example, `JMS/outQ`).
14. In the **JMS JNDI Factory Name** field, enter the factory name used to connect to the name service. This field is required. The value of com.sun.jndi.fscontext.RefFSContextFactory is the one you will probably use, if you set up your JMS configuration for WebSphere MQ as described in "Configuring the hub for the JMS transport protocol" on page 37.
15. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers" on page 228. Otherwise, click **Save**.

## Setting up a file-directory destination

### About this task

To create file-directory destinations, use the following procedure.

1. Click **Account Admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

## Destination Details

### About this task

From the Destination List page, perform the following steps:

1. Type a name to identify the destination. This is a required field.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.

## Destination configuration

### About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. In the **Address** field, enter the URI where the document will be delivered. This field is required.

   The format for UNIX and Windows systems in which the file directory is on the same drive on which WebSphere Partner Gateway is installed is: file://*<path_to_target_directory>*

   For example:

   ```
   file://localfiledir
   ```

   where *localfiledir* is a directory off the root directory.

   If File directory destination has to be created on any drive on windows, OTHER THAN the drive on which WebSphere Partner Gateway is installed, then the path is: file:///*<drive_letter>*:/*<path>*
2. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
3. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.
4. In the **Number of Threads** field, enter the number of documents that should be processed simultaneously. The default is 3.
5. In the **Validate Client IP** field, select **Yes** if you want the IP address of the sender to be validated before the document is processed. Select **No** otherwise. The default is **No**.

6. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure is about to occur because the number of retries has been exhausted. Select **No** otherwise. The default is **No**.

   When you select **Auto Queue**, all documents remain queued until the destination is placed online manually.
7. If you want the document to have its original name when it is sent to its destination, do not select **Use Unique File Name**. Otherwise, select this if you want WebSphere Partner Gateway to assign a name to the file.
8. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers" on page 228. Otherwise, click **Save**.

## Setting up an FTPS destination

### About this task

To create FTPS destinations, use the following procedure.

1. Click **Account Admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

**Note:** FTPS passive mode is not supported. See "Setting up an FTP Scripting destination" on page 223 for passive support.

## Destination Details

### About this task

From the Destination List page, perform the following steps:

1. Type a name to identify the destination. This is a required field.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.

## Destination configuration

### About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. In the **Address** field, enter the URI where the document will be delivered. This field is required.

   The format is: ftp://<*ftp_server_name*>:<*portno*>

   For example:

   `ftp://ftpserver1.ibm.com:2115`

   If you do not enter a port number, the standard FTP port is used.
2. Optionally enter a user name and password, if a user name and password are required to access the secure FTP server.

3. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
4. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.
5. In the **Number of Threads** field, enter the number of documents that should be processed simultaneously. The default is 3.
6. In the **Validate Client IP** field, select **Yes** if you want the IP address of the sender to be validated before the document is processed. Select **No** otherwise. The default is **No**.
7. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure is about to occur because the number of retries has been exhausted. Select **No** otherwise. The default is **No**.

   When you select **Auto Queue**, all documents remain queued until the destination is placed online manually.
8. In the **Connection Timeout** field, enter the number of seconds a socket will remain open with no traffic. The default is 120 seconds.
9. If you want the document to have its original name when it is sent to its destination, do not select **Use Unique File Name**. Otherwise, select this if you want WebSphere Partner Gateway to assign a name to the file.
10. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers" on page 228. Otherwise, click **Save**.

## Setting up SFTP destination

### About this task

You set up a SFTP destination so that documents can be sent from the hub to your partner's IP address. Adapter connects to the SFTP server and sends the document to the SFTP server. The document data is supplied to the adapter as a stream.

To create SFTP destinations, use the following procedure.

1. Click **Account admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

## Destination details

### About this task

In the Destination Details page, perform the following steps:

1. Enter a name to identify the destination. This is a required field.
2. Optionally, indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally, indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally, enter a description of the destination.
5. Select **SFTP** from the **Transport** list.

## Destination configuration

### About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. Enter the **SFTP host IP / Host Name**. It will accept a maximum of 100 characters. You can also enter IP addresses, IPv4, and IPv6 addresses.
2. Enter the **Port Number**. The minimum value is 1 and maximum value is 65535. The default value is 22.
3. Enter the **Output Directory**. It will accept a maximum of 100 characters. It can contain characters based on locale.
4. In the **Authentication Type**, select username/password or private key authentication.
5. In the **Auto Queue** field, select **Yes** if you want the destination to be placed offline (automatically) if a delivery failure occurs. Select **No** otherwise. The default is **No**.
6. Enter the **User Name** and **Password** for username/password. If the Authentication type is private key authentication, then enter **Username**, **Private Key File**, and **Pass Phrase**. **Private key File** is the path of the private key file in OpenSSH format.
7. Enter **Retry count**. The number of times the receiver will try to connect to the SFTP server in case the connection is not successful.
8. Enter **Retry interval**. The wait time of the receiver between retries.
9. Enter the **Number of threads**
10. The **EIS Encoding** is the encoding of the FTP server. Use this value to set the encoding for the control connection of the FTP server.
11. The **Enable server authentication** can be enabled to authenticate the server to which the connection is established. If the server authentication is enabled, enter the host key file path. The host key file has to be in OpenSSH format.
12. Click **Save** to save the configuration.
13. Enter the handler configuration and click **Save** to save the configuration details.

**Note:** Restart the corresponding server after saving the configuration:

- In simple mode, restart bcgserver server
- In simple distributed mode, restart bcgserver cluster
- In fully distributed mode, restart BCGDocMgr cluster

# Setting up an FTP Scripting destination

An FTP Scripting destination runs according to the schedule you set. The behavior of an FTP Scripting destination is governed by an FTP command script.

**Note:** If the database is down and lock User is set to "yes," the FTP Scripting destination may not work because it will not get the lock from database.

**Note:** On AIX platform, use passive mode to deliver documents with high transaction volumes. In File Transfer operation, specify passive mode in the script that is used by the FTP Scripting Destination. You can interchangeably use 'passive' or 'pasv' command in the script. The usage of active mode generates an error.

## Creating the FTP script

### About this task

To use an FTP Scripting destination, you create a file that includes all the FTP commands required that can be accepted by your FTP server.

1. Create a script for the destinations, to indicate the actions you want performed. The following script is an example of connecting to the specified FTP server (with the name and password specified), changing to the specified directory on the FTP server, and sending all the files to the specified directory on the server.

   ```
   open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD% %BCGOPTIONx%
   cd %BCGOPTIONx%
   mput *
   quit
   ```

   The placeholders (for example, %BCGSERVERIP%) are replaced when the destination is put in service by the values you enter when you create a specific instance of an FTP scripting destination, as shown in the following table:

*Table 30. How script parameters map to FTP Scripting destination field entries*

| Script parameter | FTP Scripting destination field entry |
|---|---|
| %BCGSERVERIP% | Server IP |
| %BCGUSERID% | User ID |
| %BCGPASSWORD% | Password |
| %BCGOPTIONx% | Option*x*, under **User Defined Attributes** |

   You can have up to 10 user-defined options.

2. Save the file.

## FTP script commands

You can use the following commands when creating the script:

- ascii, binary, passive, epsv

  These commands are not sent to the FTP server. They modify the mode of transfer (ascii, binary, or passive) to the FTP server.

- cd

  This command changes to the specified directory.

- delete

  This command removes a file from the FTP server.

- mkdir

  This command creates a directory on the FTP server.

- mput

  This command takes a single argument, which specifies one or more files to be transferred to the remote system. This argument can contain the standard wildcard characters to identify multiple files ('*' and '?').

- mputren

  This command takes three arguments of <source>, <temporary>, and <target> where an asterisk (*) represents the current filename being processed.

  **source** The name of the file that is being put to the FTP server. The expected value is an asterisk (*).

**temporary**
The temporary file name to use when putting the <source> to the FTP server.

**target** The file name to use for renaming the <temporary> to. After this file is renamed, the temporary file will no longer exist.

**Examples:**

**mputren * *.tmp ***
This example puts the current file to the FTP server with the extension .tmp. After putting the file to the server, the file will be renamed back to the original name.

**mputren * *.tmp *.ready**
This example puts the current file to the FTP server with the extension .tmp. After putting the file to the server, the file will be renamed back to the original with the .ready extension.

**mputren * *.tmp /complete/***
This example puts the current file to the FTP server with the extension .tmp. After putting the file to the serve, the file will be renamed back to the original but it will exist in the /complete directory. The temporary file *.tmp will not exist anymore.

**mputren * *.tmp /complete/*.final**
This example puts the current file to the FTP server with the extension .tmp. After putting the file to the server, the file will be renamed back to the original but in will exist in the /complete directory with a .final extension. The temporary file *.tmp will not exist anymore.

- open

  This command takes three parameters--the FTP server IP address, the user name, and a password. These parameters map to the %BCGSERVERIP%, %BCGUSERID%, and %BCGPASSWORD% variables.

  The first line of your FTP Scripting destination script, therefore, should be:

  ```
  open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
  ```

- quit

  This command ends an existing connection to an FTP server.

- quote

  This command indicates that everything after the QUOTE should be sent to the remote system as a command. This allows you to send commands to a remote FTP server that might not be defined in the standard FTP protocol.

- rmdir

  This command removes a directory from the FTP server.

- site

  This command can be used to issue site-specific commands to the remote system. The remote system determines if the contents of this command are valid.

# FTP Scripting destinations

## About this task

If you will be using FTP Scripting destinations, perform the following tasks:

To create FTP Scripting destinations, use the following procedure.

1. Click **Account Admin > Profiles > Partner**.

2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **Create**.

## Destination Details

### About this task

From the Destination List page, perform the following steps:

1. Type a name to identify the destination. This is a required field.
2. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
3. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
4. Optionally enter a description of the destination.

## Destination configuration

### About this task

In the **Destination Configuration** section of the page, perform the following steps:

1. Enter the IP address of the FTP server to which you are sending documents. The value you enter here will replace %BCGSERVERIP% when the FTP script is run.

   **Note:** If you are specifying an IPv6 address, provide the numeric format, not the machine name or host name.

   Examples of IPv6 addresses include:

   ```
   FEDC:BA98:7654:3210:FEDC:BA98:7654:3210
   1080:0:0:0:8:800:200C:417A
   3ffe:2a00:100:7031::1
   1080::8:800:200C:417A
   ::192.9.5.5
   ::FFFF:129.144.52.38
   2010:836B:4179::836B:4179
   ```

2. Enter the user ID and password required to access the FTP server. The values you enter here will replace %BCGUSERID% and %BCGPASSWORD% when the FTP script is run.
3. If the target is in secure mode, click **Yes** for **FTPS Mode**. Otherwise, use the default of **No**.
4. Upload the script file by following these steps:
   a. Click **Upload Script File**.
   b. Type the name of the file that contains the script for processing documents, or use **Browse** to navigate to the file.
   c. Select the **Script File Encoding Type**.
   d. Click **Load File** to load the script file into the **Currently loaded script file** text box.
   e. If the script file is the one you want to use, click **Save**.
   f. Click **Close Window**.

5. In the **Retry Count** field, enter the number of times you want the destination to attempt to send a document before it fails. The default is 3.
6. In the **Retry Interval** field, enter the amount of time the destination should wait before attempting to send the document again. The default is 300 seconds.
7. For **Connection Timeout**, enter the number of seconds a socket will remain open with no traffic. The default is 120 seconds.
8. In the **Lock User** field, indicate whether the destination will request a lock, so that no other instances of an FTP Scripting destination can gain access to the same FTP server directory at the same time.

**Note:** The **Global FTP Scripting Attributes** values are already filled in, and you cannot edit them from this page. To modify these values, you use the Global Transport Attributes page, as described in "Setting up global transport values" on page 210.

## User-defined attributes

### About this task

If you want to specify additional attributes, perform the following steps. The value you enter for the option will replace %BCGOPTION*x*% when the FTP script is run (where *x* corresponds to the number of the option.)

1. Click **New**.
2. Type a value next to **Option 1**.
3. If you have additional attributes to specify, click **New** again and type a value.
4. Repeat step 3 as often as necessary to define all the attributes.

For example, suppose your FTP script looked like this:

```
open %BCGSERVERIP% %BCGUSERID% %BCGPASSWORD%
          cd %BCGOPTION1%
          mput *
          quit
```

The %BCGOPTION% in this case would be a directory name.

## Schedule

### About this task

From the Schedule section of the page, perform the following steps:

1. Indicate whether you want interval-based scheduling or calendar-based scheduling.
   - If you select **Interval Based Scheduling**, select the number of seconds that should elapse before the destination is polled (or accept the default value).
   - If you select **Calendar Based Scheduling**, choose the type of scheduling (**Daily Schedule**, **Weekly Schedule**, or **Custom Schedule**).
     - If you select **Daily Schedule**, enter the time of day when the destination should be polled.
     - If you select **Weekly Schedule**, select one or more days of the week in addition to the time of day.
     - If you select **Custom Schedule**, select the time of day and then choose **Range** or **Selective Days** for the week and the month. With **Range**, you indicate the start date and the end date. (For example, click **Mon** and **Fri**

if you want the server polled at a certain time on weekdays only). With **Selective Days**, you choose the specific days of the week and month.

2. If you want to configure the Preprocess or Postprocess step for the destination, go to "Configuring handlers." Otherwise, click **Save**.

## Configuring handlers

### About this task

You can modify two processing points for a destination--Preprocess and Postprocess.

No handlers are provided by default for the Preprocess or Postprocess step, and, therefore, no handlers are listed by default in the **Available List**. If you have uploaded a handler, you can select it and move it to the **Configured List**.

To apply a user-written handler for these configuration points, you must first upload the handler. Refer to the *Hub Configuration Guide* for steps on uploading the handler. Then perform the following steps:

1. Select **preprocess** or **postprocess** from the **Configuration Point Handlers** list.
2. Select the handler from the **Available List** and click **Add**.
3. If you want to change the attributes of the handler, select it from the **Configured List** and click **Configure**. You will see a list of attributes that can be changed. Make the necessary changes and click **Set Values**.
4. Click **Save**.

You can further modify the **Configured List** as follows:

- Remove a handler by selecting the handler from the **Configured List** and clicking **Remove**. The handler is moved to the **Available List**.
- Rearrange the order in which the handler is processed by selecting the handler and clicking **Move Up** or **Move Down**.

## Setting up a destination for a user-defined transport

### About this task

If you want to upload a user-defined transport, perform the following steps.

1. Click **Account Admin > Profiles > Partner**.
2. Click **Destinations**.
3. Click **Manage Transport Types**.
4. Enter the name of an XML file that defines the transport (or use **Browse** to navigate to the file).
5. Use the default of **Yes** for **Commit to Database**. Select **No** if you are testing this transport before putting it into production.
6. Indicate whether this file should replace a file with the same name that is already in the database.
7. Click **Upload**.

   **Note:** From the Manage Transport Types page, you can also delete a user-defined transport type. You cannot delete a transport provided by WebSphere Partner Gateway. Also, you cannot delete a user-defined transport after it has been used for creating a destination.

8. Click **Create**
9. Type a name to identify the destination. This is a required field.
10. Optionally indicate the status of the destination. **Enabled** is the default. A destination that is enabled is ready to send documents. A destination that is disabled cannot send documents.
11. Optionally indicate whether the destination is Online or Offline. The default is **Online**.
12. Optionally enter a description of the destination.
13. Complete the fields (which will be unique for each user-defined transport) and click **Save**.

## Specifying a default destination

### About this task

After you create destinations for the internal partner or partner, select one of the destinations as the default destination.

1. Click **Account Admin > Profiles > Partner**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all partners.
3. Click the **View details** icon to display the partner's profile.
4. Click **Destinations**.
5. Click **View Default Destinations**.

   A list of destinations defined for the partner is displayed.
6. From the **Production** list, select the destination that will be the default for this partner. You can also set default destinations for other types of destinations, such as **Test**.
7. Click **Save**.

# Chapter 12. Managing connections

After you create the B2B capabilities of partners and create interactions, you establish connections between the internal partners and external partners. This chapter covers the following topics:

- "Overview of connections"
- "Activating partner connections"
- "Specifying or changing attributes" on page 232

**Note:** you should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Overview of connections

You set up a connection between partners for each type of document that will be exchanged. For example, you might have multiple connections from the internal partner to the same partner, because the packaging, protocol, document type, action, or map might be different.

When you activate connections, you can specify attributes for the source or target partner. Any attributes you set at the connection level take precedence over attributes you set at the B2B capabilities level (for a particular partner) or at the document definition level.

For EDI, XML, and ROD documents, you have multiple connections for each exchange, if the exchange involves enveloping or transformation. You can further define connections for these types of documents by selecting from a set of profiles associated with the connection. See "Connection profiles" on page 185 for details.

## Configuring multiple internal partners

WebSphere Partner Gateway does not have any restriction on the number of internal partners. It is necessary to configure the default internal partner to provide backward compatibility for Webservice and Binary documents that flow through FTPScript support features. For more information on Webservices and binary document configuration for multiple internal partners, see Chapter Configuring document types.

## Activating partner connections

### About this task

Partner connections contain the information necessary for the proper exchange of each document type. A document cannot be routed unless a connection exists between the internal partner and one of its external partners.

The system automatically creates connections between the internal partners and external partners based on their B2B capabilities and interactions.

You search for these connections and then activate them.

When selecting a Source and a Target, ensure that the source is unique.

Use the following procedure to perform a basic search for connections and then activate the connections.

1. Click **Account Admin > Connections**. The Manage Connections page is displayed.
2. Under **Source**, select a source. For example, if you are setting up an exchange that originates from the internal partner, select the Internal Partner.
3. Under **Target**, select a target. For example, if you are setting up an exchange that will be received by a partner, select that partner.

   **Note:** When you create a new connection, the Source and Target must be unique.
4. Click **Search** to find the connections that match your criteria.

   **Note:** You can also use the Advanced Search page if you want to enter more detailed search criteria.
5. To activate a connection, click **Activate**. The Manage Connections page is redisplayed, this time with the connection highlighted in green. This page shows the package, protocol, and document type for the source and target. It also provides buttons you can click to view and change partner-connection status and parameters.
6. To specify attributes for the source or target or to select a connection profile, see "Specifying or changing attributes."

For a two-action PIP, activate the connection in both directions to support the second action of the PIP. To do this, the source and target of the second action are the opposite of the source and target of the first action.

For EDI, XML, or ROD documents for which you have defined more than one interaction, make sure you activate all the connections associated with the interactions.

## Specifying or changing attributes

### About this task

When you activate the connection, you can set attributes or modify attributes that were previously defined. To specify or change the attributes for this connection:

1. Click **Attributes** to view or change the attribute values.

   For example, suppose the internal partner is sending a document packaged as None to a partner. The partner is going to receive the document packaged as AS. It is possible that the internal partner has more than one Business ID assigned to it. To indicate to WebSphere Partner Gateway which ID to use:

   a. Click **Attributes** on the Source side of the connection.
   b. When the Connection Attributes page is displayed, expand the **None** folder.
   c. Select from the **Update** list the AS ID you want sent to the partner.
   d. Click **Save**.

   **Note:** If you previously specified an AS ID (in the B2B Capabilities page, for example), the value you enter here will override the earlier value.

Another example of setting an attributes is to enter a value for the MDN address when you are receiving documents packaged as AS from a partner. The address specifies where the MDN is delivered.

2. Click **Actions** if you want to view or change an action or a transformation map associated with this connection. Any value you change here overrides any other values you have set for the action or map.
3. Click **Destinations** if you want to view or change the source or target destination.
4. If the **Add Connection Profile** button and the **Active Profiles** list appears, you can associate this connection with a particular profile that you have previously defined.

The attributes that you set at the connection level take precedence over any attributes you set at the protocol or document type level. If the attribute is associated to Package, Protocol and Document type, the value at document type will override the value set at package and protocol.

# Chapter 13. Enabling security for document exchanges

With WebSphere Partner Gateway, you can install and use several types of certificates to secure inbound and outbound transactions. This chapter includes the following topics:



Certificates and security protocols provide the following security benefits in WebSphere Partner Gateway:

- Verification as to who is sending the document
- Verification that the document has not been altered in transit
- Prevention of others from viewing the contents of the document
- Verification that the person who is sending the document is authorized to do so.

**Note:** you should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Overview of security

### Security mechanisms and protocols used in WebSphere Partner Gateway

Depending on the business protocol, WebSphere Partner Gateway uses certificates to enable these mechanisms to keep your document exchanges secure:

**Encryption and Decryption**

Encryption is a way of modifying the data so that the data is unreadable until decrypted. WebSphere Partner Gateway uses a cryptographic system known as public key encryption to secure the communication between partners and the hub. Different business protocols such as AS2 or

RosettaNet have requirements for encryption. SSL also uses encryption. In this chapter unless indicated otherwise the use of the term *encryption* applies to business protocols.

Decryption is a way of decrypting the encrypted data so that data is readable. Decryption is carried out on inbound documents.

WebSphere Partner Gateway is capable of sending an OpenPGP encrypted data. The received data package is decrypted using the private key. If the sending document is expected to be encrypted always, set the **Encryption Required** attribute to `Yes` at the target side of the connection. If the encrypted document is expected to contain Modification Detection Code packet, set the **Modification Detection** to `True` at the target side of the connection. In case, you receive an encrypted data with integrity protection, after decryption, the integrity of the data is verified using the Modification Detection Code packet. The last decrypted packet in the decrypted data has to be a Modification Detection Code Packet. In such a scenario, encrypted Data comprises of Symmetrically Encrypted Integrity Protected Data Packet, so the message integrity is verified. You have to set the encryption attributes at the target side of the connection. For the OpenPGP packet, RFC 4880 is supported. In case, you need to send encrypted data with integrity protection, set the **Modification Detection** to `True` and select the Symmetric algorithm preferences. This functionality is defined in RFC 4880 only.

**Compression**
While sending a document, in the packaging step, data has to be compressed as per the compression algorithm preference set in the connection of the target. When you receive a compressed message, it is decompressed. If the sending document is expected to be compressed always, set the **Compression Required** attribute to `Yes` at the target side of the connection. For the OpenPGP packet, RFCs 4880 is supported.

**Encryption and Compression**
When a document has to be encrypted and compressed, set all the routing object attributes for encryption and compression at the target side of the connection. The encryption is as per RFC 4880. When you receive an encrypted and compressed message, decryption is performed. After decryption, a compressed data packet is obtained on which decompression is performed. While sending encrypted data with integrity protection, set the Modification detection attribute at target side of the connection.

**Digital signature and Digital signature verification**
Digital signing is the mechanism for verifying who sent a document and that the document has not been altered in transit. It is also helpful in ensuring non-repudiation. Non-repudiation means that a partner cannot deny having originated and sent a message. It also ensures that the partner cannot deny having received a message.

**Note:** The non-repudiation information is obtained from the partner connection parameters. The partner connection parameters are obtained after a successful partner connection look-up. By default, non-repudiation is set to "Yes," which means that if the information is not available from the partner connection for some reason, the document will be put in the non-repudiation store.

**SSL** SSL is a commonly used protocol for managing security over the Internet. SSL provides secure connections by enabling two applications linked through a network connection to verify that each is trusted and by

encrypting the data to ensure confidentiality. The encryption is independent of the data type. SSL is used over transports such as HTTP and FTP.

**Basic authentication**

When any incoming message is sent over HTTP or HTTPS, the receiver can authenticate the sending partner by the basic authentication credentials. The user ID and password are passed in the HTTP header. As password is also sent, Basic authentication should be used with SSL/TLS to ensure that the headers are encrypted. The authentication is provided either using Bussiness ID/username:password or Username:password in Base64 encoded format. The value in the HTTP header is considered only if **Enable basic authentication** is set to true. Select Basic Authentication in the Receiver details page of the console to set it to True.

If authentication fails, the Authentication failed response is returned to the sender. Otherwise, the document is sent for further processing. In case of SSL client authentication, the business ID(s) of the sending partner is identified. When the document is received, the receiver checks if the certificate is associated with any partner, otherwise the document fails if there is no match. For backward compatibility, while sending a SOAP message with basic authentication, set the **Enable Basic Authentication** flag to "No" at the receiver. Unless the authentication of the document fails at the receiver, you can view the document in the document viewer. Basic Authentication is supported for the following documents:

- EDI/XML documents
- AS2 documents with binary/EDI/XML Payload
- Webservices request
- Rosettanet message
- ebMS message

Security can either be at the transport or business protocol. The authentication of users at the receiver supports binary documents from external partners over HTTP. The sending partner is identified either using basic authentication credentials or using SSL client authentication credentials.

## Certificates and security mechanisms

Certificates form the basis of all three approaches to security: encryption, digital signatures, and SSL. They enable these approaches in WebSphere Partner Gateway. Using a certificate helps keep documents secure during transmission.

Each partner has one or more certificates for sending or receiving documents with WebSphere Partner Gateway, and WebSphere Partner Gateway represented by the Hub Operator has a one or more certificates for sending or receiving documents with the partner.

**Note:** The same certificates used for a partner or the Hub Operator apply to all documents. Certificates are not varied by document type.

### Certificates and encryption

A certificate contains the public key part of a mathematically related public/private key pair. The public key “locks” or encrypts a document before it is sent and does so in such a way that only the private key can then “unlock” or decrypt a document after it is sent. A public key is called a public key because you share it with partners who send you encrypted documents while the private key is

kept to yourself so you can decrypt them. A certificate contains the public key and binds it to a Subject Name, which is the name for the End-Entity to whom the certificate belongs.

Certificates are generated by the partner and are either self-signed by the partner or CA-issued. A CA-issued certificate is a certificate that a partner has requested using a Certificate Signing Request (CSR) and received from a certifying authority (CA). A CA-issued certificate is signed by the CA and not by the partner. Each partner has at least one certificate to use in sending or receiving documents.

Business document encryption only applies if the business standard supports encryption. Not all standards support encryption. For the standards that do support encryption each standard has different ways for applying the encryption. WebSphere Partner Gateway understands the differences between the standards and how to apply the encryption.

If WebSphere Partner Gateway is sending a document to a partner then that partners certificate is used to encrypt the document. This way only the partner can read the contents by decrypting the document with their own private key. The certificate that is used will be the encryption certificate loaded into WebSphere Partner Gateway for that partner.

If a partner is sending a document to WebSphere Partner Gateway then the partner uses the Hub Operator's certificate to encrypt the document. This way only the Hub Operator who has the private key can read the contents by decrypting the document. The private key that is used is the one loaded for the Hub Operator under the Load PKCS12 option. Note that the Hub Operator's certificate has to be given to the partner by the administrator.

**Notes:**

1. WebSphere Partner Gateway supports the RC2 and TripleDES algorithms. It does not support the RC5 algorithm. If you were using the RC5 algorithm in an earlier release, switch to one of the supported algorithms.
2. WebSphere Partner Gateway also supports the following algorithms:
   - AES, TripleDES, and RC2: for sent and received ebMS documents.
   - TripleDES and RC2: for RNIF documents.
   - DES: for ebMS, but it is recommended to use stronger algorithms like RC2, TripleDES, or AES.

   You can set these algorithms in the WebSphere Partner Gateway Console System Administration > DocMgr Administration > Security view or with the SecurityService API in User Exits. Refer to the *WebSphere Partner Gateway Administrator Guide* for information about the Security properties. Refer to the *WebSphere Partner Gateway Programmer Guide* for information about SecurityService.

## Basic procedure

To receive an encrypted document, you must complete the following basic steps. For the complete procedure, see "Using certificates to enable encryption and decryption" on page 246.

1. Obtain a public/private key pair either by generating it yourself or by receiving one from a CA.

2. Upload the private key to your WebSphere Partner Gateway server under the Hub Operator (key can be used by all the internal partners) or Internal partner (the key can be used only by that specific internal partner), so that the server can decrypt incoming documents.
3. Provide the public certificate to your trading partner so that your partner can upload the certificate into that partner's server and that partner can encrypt documents before sending them to you.

Once you have completed this procedure, this partner, using your certificate can send you documents that are encrypted in such a way that only you can decrypt them. To send partners encrypted documents, you must reverse this procedure, uploading their certificates and using those certificates to encrypt documents to send to them.

## Certificates and digital signing

WebSphere Partner Gateway supports digital signature as required by the B2B protocols.You use certificates for signing similar to the way you use encryption certificates except that it is reversed. You must create the certificate to send a document with a digital signature to partners, not vice versa.

Digital signatures are used to verify the actual sender of the document and to prove that the document has not been altered in transit. They only apply if the business standard supports digital signatures. Not all standards support digital signatures. For the standards that do support digital signatures each standard has different ways for applying the digital signatures. WebSphere Partner Gateway understands the differences between the standards and how to apply the digital signatures.

If WebSphere Partner Gateway is sending a document to a partner then the Hub Operators private key loaded under the Load PKCS12 option is used to sign the document. The partner uses the Hub Operators certificate to verify that WebSphere Partner Gateway is the one that signed the document. If the Hub Operator's private key was not used to sign the document then the Hub Operator's certificate that the partner has will not work to verify the signatures. Note that the Hub Operator's certificate has to be given to the partner by the administrator.

If a partner is sending a document to WebSphere Partner Gateway then WebSphere Partner Gateway uses the partner's Digital Signature certificate to verify that the partner is the one that signed the document. If the partner's private key was not used to sign the document then the certificate that WebSphere Partner Gateway has for that partner will not work to verify the signature.

### Basic procedure:

To send a digitally signed document, you must complete the following basic steps. For the complete procedure, see “Using certificates to enable digital signing” on page 251.

1. Obtain a public/private key pair either by generating it yourself or by receiving one from a CA.
2. Upload the private key to your WebSphere Partner Gateway server under the Hub Operator so that the server can sign the documents being sent.
3. Provide the public certificate to your trading partner so that that partner can upload the certificate into that partner's server and that partner can verify documents that are received from you.

Once you have completed this procedure, you, using your private key, can send documents that are digitally signed so that the partner knows that no one else could have sent them. To receive similarly signed documents from partners, you must reverse this procedure, uploading their certificates and using them to ascertain their origin.

## Certificates and SSL/TLS

When sending documents, you can use SSL to encrypt documents so that only the recipient can read those documents, therefore ensuring data confidentiality.

Within SSL is the notion of a *client* and a *server*. A client connects to a server in order to send a document to the server. When the client connects to the server, the server will send the client a certificate to use in encrypting the document. This server certificate is also part of server authentication, which means that the server uses its certificate to authenticate itself to clients. Sometimes the server will also request a certificate from the client. This is called Client Authentication and is used by the server to verify that the client is known to the server.

When WebSphere Partner Gateway is sending a document to a partner, WebSphere Partner Gateway is the client, and the partner is the server (meaning, the document is being sent to the partner's server).

**Note:** The partner's server is the destination defined in WebSphere Partner Gateway for this partner.

When the partner is sending a document to WebSphere Partner Gateway, then the partner is the client and WebSphere Partner Gateway is the server.

**Note:** This is the receiver that has been defined in WebSphere Partner Gateway.

When a partner is sending a document to WebSphere Partner Gateway using SSL, the actual identity of the partner is not known. If Client Authentication is being used, the identity of the partner is still not known. However, what is known is that this partner is trusted for sending documents to WebSphere Partner Gateway. WebSphere Partner Gateway also has an additional feature for identifying the partner from the Client Authentication certificate that the partner provided.

If WebSphere Partner Gateway is sending a document to a partner, then that partner's certificate is used to encrypt the document. Therefore, only that partner can read the contents by decrypting the document with that partner's own private key. As part of SSL during runtime, the partner will dynamically send the certificate to use for encryption to WebSphere Partner Gateway. WebSphere Partner Gateway verifies that the certificate is valid by building and validating the certification path using the certificates loaded as Root/Intermediate certificates under the Hub Operator.

There is a second optional part of SSL called Client Authentication for validating the sender in which the partner requests a certificate from WebSphere Partner Gateway. WebSphere Partner Gateway will send the Client Authentication certificate that was loaded under the Hub Operator. Note that the Hub Operator's certificate for Client Authentication has to be given to the partner by the administrator. If the Client Authentication certificate is self-signed, then the self-signed certificate needs to be given to the partner. If the Client Authentication certificate is CA-issued, then the CA certificate may need to be given to the partner, if the partner does not already have the CA certificate.

If a *partner* is sending a document to WebSphere Partner Gateway using SSL, the WebSphere Partner Gateway certificate is used to encrypt the document. Therefore, only WebSphere Partner Gateway can read the contents by decrypting the document with its own private key. As part of SSL during runtime, WebSphere Partner Gateway will dynamically send the certificate to use for encryption to the partner. The partner verifies that the certificate is valid by comparing it with the certificate that the Administrator has previously given the partner. There is a second optional part of SSL called Client Authentication for validating the sender in which the WebSphere Partner Gateway requests a certificate from the partner. The partner will send the Client Authentication certificate to WebSphere Partner Gateway and this certificate will be verified against the certificate that the partner has previously given the administrator.

**Note:** For receiving document from partners using SSL, WebSphere Partner Gateway uses the underlying WebSphere Application Server facilities. Therefore, the certificates used during runtime are not uploaded using the WebSphere Partner Gateway Console but are instead loaded into the WebSphere Application Server key store and trust store.

With Client Authentication there is an additional partner identification that WebSphere Partner Gateway performs outside of the SSL transport. The Client Authentication certificate that was provided by the partner will be passed to WebSphere Partner Gateway who will compare this to the certificate loaded for that partner's SSL Client so that the partner can be identified.

An HTTP-based SSL connection is always initiated by the client using a URL starting with https:// instead of http://. An SSL connection begins with a handshake. During this stage, the applications exchange certificates, agree on the encryption algorithms to use, and generate encryption keys used for the remainder of the session.

### Basic procedures

To *send* a document using SSL, you must complete the following basic steps. For the complete procedure, see "Using certificates to enable SSL" on page 255.

1. Obtain a certificate from your partner and load into the WebSphere Application Server Trust store.
2. For Client Authentication to the partner obtain a public/private key pair either by generating it yourself or by receiving one from a CA.
3. Upload the private key and public certificate to your WebSphere Application Server Key store.
4. Provide the public certificate to your trading partner so that partner can upload the certificate into that partner's server and that partner can verify the Client Authentication certificate is received from you during the SSL runtime communication.

To *receive* a document using SSL, you must complete the following basic steps. For the complete procedure, see "Using certificates to enable SSL" on page 255.

1. Obtain a public/private key pair either by generating it yourself or by receiving one from a CA.
2. Upload the private key and public certificate to your WebSphere Application Server Key store.

3. Provide the public certificate to your trading partner so that partner can upload the certificate into that partner's server and that partner can verify the Server certificate is received from you during SSL runtime communication.
4. For Client Authentication obtain a certificate from your partner and load into the WebSphere Application Server Trust store. This will be used during the SSL runtime communication.
5. For identifying the partner from the Client Authentication certificate in the WebSphere Partner Gateway Console upload the partners certificate under the partner's Client Authentication.

## Storing certificates in key stores and trust stores

WebSphere Partner Gateway has two ways of storing certificates. For documents being sent by a partner to WebSphere Partner Gateway using SSL, certificates are stored in the WebSphere Application Server key store and trust store. Trust stores are used to store certificates that are trusted which in turn are used to validate that a certificate received from a partner is valid. Key stores are used to store the public and private key of the WebSphere Partner Gateway Hub Operator. Certificates being used for business document security are stored by loading through the WebSphere Partner Gateway Console. This section describes the key store and trust store used with WebSphere Application Server. When you install WebSphere Partner Gateway, a key store and a trust store are created for the WebSphere Application Server that the Receiver and Console are installed on.

- A key store is a file that contains your public and private keys.
- A trust store is a key database file that contains the public keys for your partners' self-signed and CA certificates. The public key is stored as a signer certificate. For commercial CA, the CA root certificate is added. Because the trust store file does not contain your private key the trust store file can be more publicly accessible than the key store file.
- iKeyman is used for administering the key store and trust store. This utility is described in the sections that require its use.

  **Note:** WebSphere Application Server administrative console can also be used to manage certificates, key stores, and trust stores for the Receiver and Console. See the article entitled "Securing applications and their environment" in the WebSphere Application Server Information Center for details on how to manage certificates and key stores using the WebSphere Application Server administrative console.

By default, a key store and trust store are created in the *<ProductDir>*/common/security/keystore directory. The names are:

- bcgSecurity.jks
- bcgSecurityTrust.jks

### Changing the default password

The default password for accessing the stores is WebAS. The WebSphere Application Server is configured to use these stores. You can use the iKeyman utility to change the password. Alternatively, you can use the keytool command to change the password of the key store file. In UNIX, the command will be as follows:

```
/<WAS_Installation_Dir>/java/bin/keytool
 -storepasswd -new $NEW_PASSWORD$ -keystore $KEYSTORE_LOCATION$
 -storepass $CURRENT_PASSWORD$ -storetype JKS
```

In Windows, use the preceding command but use backslashes and drive names instead.

If the key store passwords are changed, each WebSphere Application Server instance configuration must also be changed. This can be done using the `bcgChgPassword.jacl` script. For the Console instance, navigate to the following directory:

`/<ProductDir>/bin`

and issue the following command:

```
./bcgwsadmin.sh -f /<ProductDir>/scripts/
bcgChgPassword.jacl -conntype NONE
```

Repeat this command for the WebSphere Application Server instances of the Receiver and Document Manager.

**Note:** For Windows installations, use `bcgwsadmin.bat` instead of `./bcgwsadmin.sh`.

You will be prompted for the new password.

### Replacing an expired certificate

If a certificate in a trust store has expired, you must add a new certificate to replace it by using the following procedure:

1. Start iKeyman, if it is not already running.
2. Open the trust store file.
3. Type the password and click **OK**.
4. Select **Signer Certificates** from the menu.
5. Click **Add**.
6. Click **Data type** and select a data type, such as Base64-encoded ASCII data.
   This data type must match the data type of the importing certificate.
7. Type a certificate file name and location for the CA root digital certificate or click **Browse** to select the name and location.
8. Click **OK**.
9. Type a label for the importing certificate.
10. Click **OK**.

## Using certificate chains

A certificate chain is made up of a partner's certificate and any certificates used to authenticate the partner's certificate. For example, if a CA was used to create the partner's certificate, that CA might itself have been certified by another CA. The chain of trust begins at the *root* CA (the trust anchor). The root CA's digital certificate is self-signed; that is, the certificate authority uses its own private key to sign the digital certificate. Any certificates between the trust anchor and the partner's certificate (the target certificate) are *intermediate* certificates.

For any CA-issued certificates, all certificates in the chain must be added. For example, in a certificate chain in which A (the trust anchor) is the issuer of B and B is the issuer of C (the target certificate), certificates A and B must be uploaded as CA certificates.

WebSphere Partner Gateway treats all self-signed certificates as trust anchors. The self-signed certificate can be of a certifying authority (CA), or it can be a self-signed certificate generated by the partner.

For inbound SSL, all root (trust anchor) certificates and intermediate certificates are kept in the WebSphere Application Server Trust store as described earlier. For all partners certificates that their root (trust anchor) certificates and intermediate certificates are uploaded under the Hub Operator.

## Using primary and secondary certificates

You can create more than one certificate of a particular type and designate one as the primary certificate and one as the secondary certificate. If the primary certificate expires or is otherwise unable to be used, WebSphere Partner Gateway switches to the secondary certificate.

**Note:** This feature can be used to transition from old certificate to a new certificate without stopping the server.
You specify, on the Community Console, which certificate is primary and which is secondary.

The ability to provide primary and secondary certificates is available for the following certificates:

- Encryption certificate of a partner
- Signing certificate of the Hub Operator
- SSL Client certificate of the Hub Operator

## Changing the cryptographic strength

The Java Runtime Environment (JRE) that ships with WebSphere Partner Gateway enforces restrictions regarding the cryptographic algorithms and maximum cryptographic strengths available for use. For example, restricted policy specifies limits on the allowable length, and, as a result, strength of encryption keys. These restrictions are specified in files called *jurisdiction policy files*. The maximum allowable length is 2048 bytes.

If you want to support certificates with a key size greater than 2048 bytes, use the unrestricted or unlimited strength version of the jurisdiction policy files. You can specify that you want to use stronger, unrestricted policy by installing new policy files to a subdirectory of the installed JRE.

There are also encryption restrictions on the symmetric key algorithms, such as 3DES. If you need a strong symmetric key algorithm, replacing the jurisdiction policy files will also remove the restrictions for the symmetric keys. For example, if you are using AES algorithm, then unrestricted cryptography policy files are required. Refer to the link http://www.ibm.com/developerworks/java/jdk/security/50 for details.

However, due to import control restrictions, the jurisdiction policy files shipped with the IBM SDK for Java 5 Development Kit allow **strong** but limited cryptography to be used. The following table provides the maximum key sizes allowed by this **strong** version of the jurisdiction policy files:

*Table 31. Maximum key size of algorithms used in strong jurisdiction policy files*

| Algorithm | Maximum Key Size |
|---|---|
| DES | 64 |

*Table 31. Maximum key size of algorithms used in strong jurisdiction policy files (continued)*

| Algorithm | Maximum Key Size |
|---|---|
| DESede | 112 (effective) or 168 (effective) |
| RC2 | 128 |
| RSA | 2048 |
| * (all others) | 128 |

**Note:** An 'Encryption failure XMLEncryptionException' exception occurs while encrypting a routed ebMS message having the following parameters:

- Encryption Algorithm :aes-192-cbc or aes-256-cbc
- Encryption Protocol : Xml Encryption

To resolve this issue, if legally permitted, install unrestricted cryptography policy files.

### Installation instructions for the Windows, Linux and AIX operating systems

To install unlimited jurisdiction policy files in WebSphere Partner Gateway, perform the following steps:

1. Download the unlimited jurisdiction strength policy files from the **IBM SDK Policy files** link at the following Web site: http://www.ibm.com/developerworks/java/jdk/security/50/.
2. Unzip the downloaded file to a temporary folder.
3. Copy local_policy.jar and US_export_policy.jar from the temporary folder.
4. Stop all servers that are hosted by the instance of WebSphere Application Server that you are configuring.
5. Change to the folder <*WASInstallationDir*>\java\jre\lib\security.
6. Rename the existing local_policy.jar and US_export_policy.jar to local_policy.jar.bak and US_export_policy.jar.bak.
7. Paste the jar files copied in step 3 to the folder <*WASInstallationDir*>\was\java\jre\lib\security.
8. Restart the servers that are hosted by the instance of WebSphere Application Server that you have just reconfigured.

These steps apply to all the WebSphere Application Server installations in which WebSphere Partner Gateway applications are installed.

### Installation instructions for the HP-UX and Solaris operating systems

For HP-UX and Solaris platforms, the following instructions are applicable:

1. Download the unlimited jurisdiction strength policy files from the **IBM SDK Policy files** link at the following Web site: http://www.ibm.com/developerworks/java/jdk/security/50/.
2. Unzip the downloaded file to a temporary folder.
3. Stop all servers that are hosted by the instance of WebSphere Application Server that you are configuring.
4. Change to the folder <*WASInstallationDir*>\java\jre\lib\security.
5. Rename the existing local_policy.jar and US_export_policy.jar to local_policy.jar.bak and US_export_policy.jar.bak.

6. Copy local_policy.jar and US_export_policy.jar from the temporary folder to <*WASInstallationDir*>\java\jre\lib\security folder.
7. Restart the servers that are hosted by the instance of WebSphere Application Server that you have just reconfigured.

These steps apply to all the WebSphere Application Server installations in which WebSphere Partner Gateway applications are installed.

### SSL with Client Authentication configuration

If you will be sending documents using a transport protocol with SSL with Client Authentication, then an additional change needs to be made for the JSSE provider that is used. See, Chapter 14, "Troubleshooting "SSL handshake fails due to no certificate received," in the *WebSphere Partner GatewayAdministrator Guide* for additional information.

### Certificate expiration

Only certificates that are used for encryption, digital signature, and SSL are disabled when they expire. These certificates should be end-entity certificates and not CA certificates. CA certificates are not disabled when they expire.

If the root or intermediate certificates expire between server restarts, they will not be included in the list of trusted certificates. This means that if the certification path build fails because the CA certificate was not found, it could be that the CA certificate has expired. If a root or intermediate certificate expired in runtime, the certification path build will fail and the corresponding end-entity certificate will not be used in the business transaction. You can check the validity period and status of the certificate using the Certificate List view in the WebSphere Partner Gateway Console. The validity date of the expired certificates appear red in this view.

If a CA certificate has expired, you can obtain a new certificate from the CA that issued it. Upload the new CA certificate using the WebSphere Partner Gateway Console. See “Using certificates to enable encryption and decryption,” “Using certificates to enable digital signing” on page 251, and “Using certificates to enable SSL” on page 255 for information on uploading certificates.

## Using certificates to enable encryption and decryption

This section describes encryption and decryption of certificates.

### Creating and installing inbound decryption certificates

This certificate is used by the hub to decrypt encrypted files received from partners. The hub uses your private key to decrypt the documents. Encryption is used to keep anyone other than the sender and intended recipient from viewing documents in transit.

Note the following important restriction about receiving encrypted AS2 messages from partners. If a partner sends an encrypted AS2 message but uses the wrong certificate, the decryption fails. No MDN is returned to the partner to indicate the failure, however. In order for your partner to receive MDNs in this situation, create a connection to the partner with the following document definition:

- Package: **AS** to Package: **None**
- Protocol: **Binary**to Protocol: **Binary**
- Document Type: **Binary**to Document Type: **Binary**

The connection created must be AS to None connection, that is, creating a connection by activating the AS B2B capability on one partner and None B2B capability on the other. Please ensure that the source gateway on the AS side is a SMTP gateway (in case of AS1), HTTP gateway (in case of AS2) or FTP gateway (in case of AS3), which is configured to MDN address. Thus, the decryption failure MDN is sent back over this AS to None Binary connection.

## Step 1: Obtain a certificate

### About this task

**Generating a self-signed certificate:** If you are going to use decryption, use the following procedure.

1. Start the iKeyman utility.
2. Use iKeyman to generate a self-signed certificate and a key pair.
3. Use iKeyman to extract to a file the certificate that will contain your public key.
4. Distribute the certificate to your partners. They are required to import the file into their B2B product for use as an encryption certificate. Advise them to use it when they want to send encrypted files to the internal partner. If your certificate is CA-signed, provide the CA certificate as well.
5. Use iKeyman to save the self-signed certificate and private key pair in the form of a PKCS12 file.
6. Navigate to **Profile** > **{Hub Operator/internal partner}** > **certificates** > **Load certificate.**
7. In the **Which Partner does this Certificate(s) belong to** drop-down, select the partner to associate the newly uploaded Certificate.
8. Click **Search** to find specific or sub-set of partners.
9. Click **Browse** next to **Certificate Location** to upload the Certificate.
10. Click **Next**.
11. In the Provide certificate details, enter the following certificate information: **Leaf certificate**, **Root CA certificate** Or **intermediate CA certificate**.
12. Associate this certificate to **Decryption**.
13. In the **Certificate usage**, select **Primary** or **Secondary**.
14. Select **enabled** or **disabled** in the **Status** based on whether you want to enable or disable the Certificate after uploading
15. Select the **Operation mode**.
16. Click **Finish** to save the changes and close the wizard.

**Using a CA-signed certificate:** If you are going to use a certificate signed by a CA, use the following procedure:

1. Start the iKeyman utility.
2. Use iKeyman to generate a certificate request and a key pair for the Receiver.
3. Submit a Certificate Signing Request (CSR) to a CA.
4. When you receive the signed certificate from the CA, use iKeyman to place the signed certificate into the key store.

## Step 2: Distribute the certificate

### About this task

Distribute the signing CA certificate to all partners.

# Installing outbound encryption certificates

The outbound encryption certificate is used when the hub sends encrypted documents to partners. WebSphere Partner Gateway encrypts documents with the public keys of the partners, and the partners decrypt the documents with their private keys.

The partner can have more than one encryption certificate. One is the primary certificate, which is the one used by default. The other is a secondary certificate, which is used if the primary certificate expires.

## Step 1: Obtain a partner's certificate

### About this task

Obtain the partner's encryption certificate. The certificate must be in X.509 DER format. Note that WebSphere Partner Gateway supports only X5.09 certificates.

## Step 2: Install the partner's certificate

### About this task

Install the certificate through the Community Console under the partner's profile by completing the following procedure:

1. Navigate to **Profile** > **External partner** > **certificates** > **Load Certificate**.
2. In the **Select Partner, File Location, Password** page of the wizard, enter the following values:
   - **Which partner does this certificate(s) belongs to**: Select the partner to associate the newly uploaded certificate. Click Search to find a specific partner or subset of a partners. If the partner is a Hub Operator or Internal Partner, enter the certificate location, private key location, and password (OR) Provide the truststore or keystore with password. For External Partner, enter the certificate location (OR) provide the trust store location containing the certificate chain.
   - **Certificate Location**: Click **Browse** to select the location of the certificate public.
3. Click **Next** to go to the **Certificate Details** page of the wizard.
4. In the **Certificate Details** page of the wizard, enter the following details of the certificate:
   - **Leaf Certificate Name** - The name of the Leaf Certificate. The field name depends on whether the certificate is a Leaf certificate, Root CA certificate or an intermediate CA certificate.
   - **Description** - The description of the Leaf Certificate.
   - **Certificate Type** - Associate this certificate to Encryption.
   - **Certificate Usage** - Associate an usage for the certificate. The values are Primary and Secondary.
   - **Operation Mode** - Enter the mode of operation.
   - **Status** - Select enabled or disabled based on whether you want to enable or disable a certificate after upload. The Next button is enabled only if the certificate is enabled.
   - **Set Management** - You can either associate a certificate to an existing set or create a new set. If the certificate is a secondary certificate, it can only be associated to an existing set. You can associate the certificate to any set for an internal partner with type encrypt or for a external partner with type SSL (Incoming client auth) or Signing (Verify).

5. Click **Next** to go to Set page of the wizard. If the certificate is primary, you do not have to create sets and associate the certificate to a set and participant connection. If you have selected **Create new set** check box, then **Create New Set** page of the wizard will open. Otherwise, the **Add to Existing** page of the wizard will open. If the file contains a private key of the internal partner or the public certificate of the external partner used for SSL / Digital Signature, then you can click **Finish**.
6. In the **Create New Set** page of the wizard, enter the details of the new set. For Primary certificates, you do not have to create sets and associate a certificate to it. Enter the following values:
   - **Set Name** - The name of the Set.
   - **Description** - The description of the Set.
   - **Status** - Select enabled or disabled. If it is disabled the **Next** button will not be enabled.
   - **Make default settings** - Select this check box if you want this set to be the default.
7. In the **Add to Existing Set** page of the wizard, select set(s) to add the certificate. Enter the following values:
   - **Select from the list of Sets available for the selected Certificate type** - From the list, select set(s) to add the certificate.
   - **Make default settings** - Select this check box if you want this set to be the default.
8. From the **Create New Set** or **Add to Existing Set**, click **Next** to go to the **Default Settings** page of the wizard. The **Next** button will be enabled only if the status of the set is enabled.
9. Select **enabled** or **disabled** in the **Status** based on whether you want to enable or disable the Certificate after upload.

   **Note:** If you have selected the **Make default set** check box in the earlier page (Create new set or Add to exisiting set), then you need to associate the set to an operation mode. This will display certificate usages against operation modes. The encryption will be disabled for internal partners. SSL Client and Digital Signature will be disabled for external partners.
10. Click **Next** to go to the Configuration page of the wizard. In case you click **Finish** and there are some missing roots or intermediate CA certificates, you will be prompted to upload. If you click "Yes" in the prompt window, the first page of the wizard will open. Click **Cancel** if you want to upload at a later stage.
11. In the Configuration page of the wizard, enter the following values:

    **Note:** The Configuration page displays a list of certificate(set) usage against operation modes. The current set name is pre-populated for all, but you can reset it.
    - **From Partner** - This field will be pre-populated with the value of the internal partner.
    - **To Partner** - This drop-down is pre-populated with the list of all external partners. You can also select the value "All" to include all external partners.
    - **From Package** - From the drop-down, select the package Document Flow Definitions objects of the internal partner.
    - **To Package** - From the list, select the package Document Flow Definitions objects of the external partner.

12. Click **Add more connections** if you want to associate the set to other participant connections.
13. Click **Add Secondary Certificate** to add a secondary certificate to the current set.
14. Click **Finish** to upload the Certificate. In case there are some missing roots or intermediate CA certificates, you will be prompted to upload. If you click "Yes" in the prompt window, the first page of the wizard will open. Click **Cancel** in the prompt window if you want to upload at a later stage.

Repeat this step if the partner has a second encryption certificate.

## Step 3: Install any CA-issued certificates
### About this task

If the certificate was signed by a CA and the CA root certificate and any other certificates that are part of the certificate chain are not already installed in the Hub Operator profile, install the certificates now by following this procedure:

**Note:** You do not have to perform this step if the CA-issued certificate is already installed.

1. Navigate to **Profile** > **<Hub Operator> user**> **certificates** > **Load Certificate.**
2. In the **Which Partner does this Certificate(s) belong to** drop-down, select the partner to associate the newly uploaded Certificate.
3. Click **Search** to find specific or sub-set of partners.
4. Click **Browse** next to **Trust store (or) Keystore location**.
5. For both Certificate and Trust store, enter **Password**.
6. If Trust store, enter the **Keystore type** click **Next**.
7. In the **Select end entity certificate to upload** page of the wizard, select a certificate to be loaded.

   **Note:** When you load certificates using a trust store that has more than one certificate, the **Select the list of root and intermediate CA Certificates to be uploaded** is populated with all the certificates. You can also upload multiple certificates.
8. Click **Finish**.

## Step 4: Enable encryption
### About this task

Enable encryption at the package (highest level), partner, or connection level (lowest level). Your setting can override other settings at the connection level. The connection summary will inform you if any required attribute is missing.

For example, to alter the attributes of a partner connection, click **Account Admin > Connections> Partner Connections** and then select the partners. Click **Attributes** and then edit the attribute (for example, **AS Encrypted**).

When the error message `No valid encryption certificate found` is displayed, neither the primary nor the secondary certificate is valid. The certificates might be expired or they might have been revoked. If the certificates were expired or revoked, the corresponding event (`Certificate revoked or expired`) can also be seen in the Event Viewer. Note that these two events might be separated by other events.

To display the Event Viewer complete the following:

1. Click **Viewers > Event Viewer**.
2. Select the appropriate search criteria.
3. Click **Search**.

See the *WebSphere Partner Gateway Administrator Guide* for information on using the Event Viewer.

# Using certificates to enable digital signing

## Creating an outbound signature certificate

The Document Manager uses this certificate when it sends outbound, signed documents to partners. The same certificate and key are used for all ports and protocols.

You can have more than one digital signature certificate. One is the primary certificate, which is the one used by default. The other is a secondary certificate, which is used if the primary certificate expires.

### Generating a self-signed certificate

#### About this task

If you are going to use a self-signed certificate, use the following procedure.

1. Start the iKeyman utility.
2. Use iKeyman to generate a self-signed certificate and a key pair.
3. Use iKeyman to extract to a file the certificate that will contain your public key.
4. Distribute the certificate to your partners. The preferred method for distribution is to send the certificate in a zipped file that is password protected, by e-mail. Your partners must call you and request the password for the zipped file.
5. Use iKeyman to export the self-signed certificate and private key pair in the form of a PKCS12 file.

### Installing outbound self-signed certificate

#### About this task

1. Navigate to **Profile** > {**Hub Operator/Internal partner**} > **certificates** > **Load Certificate**.
2. In the **Select Partner, File Location, Password** page of the wizard, enter the following values:
   - **Which partner does this certificate(s) belongs to**: Select the partner to associate the newly uploaded certificate. Click **Search** to find a specific partner or subset of a partners. If the partner is a Hub Operator or Internal Partner, enter the certificate location, private key location, and password (OR) Provide the truststore or keystore with password. For External Partner, enter the certificate location (OR) provide the trust store location containing the certificate chain.
   - **Private Key:** Click **Browse** to select the Private Key of the certificate.
   - **Password:** If the certificate has a password, enter the value.
   - **Trust Store (or) Keystore Location:** Click **Browse** to select the Keystore Location. Key store is a collection of private keys along with trusted root and CA certificates.
   - **Password:** Enter the password for Keystore Location.

- **Type:** Select the type of Trust store (or) Keystore. The available values in the drop-down are: JKS, JCEKS, and PKCS12.

**Note:** In Web Sphere Partner Gateway, on creating a CMS type key database (keystore) with iKeyman, the following error was displayed:

```
The CMS java native library was not found. Please make sure the
 SSL component required by your product is installed and library path is defined properly
```

. As WebSphere Application Server and Web Sphere Partner Gateway do not use CMS keystores, use supported keystore type, JKS (default), PKCS12, or JCEKS.

3. Click **Next** to go to **Certificate Details** page of the wizard. The **Select end entity and CA certificates** page of the wizard will open when you load certificates via a trust store that has more than one certificate. The list of certificates available in the trust store is displayed.
4. In the **Select end entity certificate and CA Certificate** page of the wizard, enter the following values:
   - **The keystore contains more than one End Entity certificate. Select the certificate to be uploaded?** - The drop-down has a list of all the End Entity certificates. Select the certificate to upload.
   - **Password** - If the keystore has a password, select this check box and enter the password in the text box.
   - **Select the List of Root and Intermediate CA certificates to be uploaded** - From the list box, select the Root and Intermediate CA certificates to upload.
5. Click **Next** to go to the **Certificate Details** page of the wizard.
6. In the **Certificate Details** page of the wizard, enter the following details of the certificate:
   - **Leaf Certificate Name** - The name of the Leaf Certificate. The field name depends on whether the certificate is a Leaf certificate, Root CA certificate or an intermediate CA certificate.
   - **Description** - The description of the Leaf Certificate.
   - **Certificate Type** - Associate this certificate to Encryption.
   - **Certificate Usage** - Associate an usage for the certificate. The values are Primary and Secondary.
   - **Operation Mode** - Enter the mode of operation.
   - **Status** - Select enabled or disabled based on whether you want to enable or disable a certificate after upload. The Next button is enabled only if the certificate is enabled.
   - **Set Management** - You can either associate a certificate to an existing set or create a new set. If the certificate is a secondary certificate, it can only be associated to an existing set. You can associate the certificate to any set for an internal partner with type encrypt or for a external partner with type SSL (Incoming client auth) or Signing (Verify).

     **Note:** For hub operator, there will not be any set management. The certificates will be associated to the default set created.
7. Click **Next** to go to Set page of the wizard. If the certificate is primary, you do not have to create sets and associate the certificate to a set and participant connection. If you have selected **Create new set** check box, then **Create New Set** page of the wizard will open. Otherwise, the **Add to Existing** page of the

wizard will open. If the file contains a private key of the internal partner or the public certificate of the external partner used for SSL / Digital Signature, then you can click **Finish**.

8. In the **Create New Set** page of the wizard, enter the details of the new set. For Primary certificates, you do not have to create sets and associate a certificate to it. Enter the following values:
   - **Set Name** - The name of the Set.
   - **Description** - The description of the Set.
   - **Status** - Select enabled or disabled. If it is disabled the **Next** button will not be enabled.
   - **Make default settings** - Select this check box if you want this set to be the default.
9. In the **Add to Existing Set** page of the wizard, select set(s) to add the certificate. Enter the following values:
   - **Select from the list of Sets available for the selected Certificate type** - From the list, select set(s) to add the certificate.
   - **Make default settings** - Select this check box if you want this set to be the default.
10. From the **Create New Set** or **Add to Existing Set**, click **Next** to go to the **Default Settings** page of the wizard. The **Next** button will be enabled only if the status of the set is enabled.
11. Select **enabled** or **disabled** in the **Status** based on whether you want to enable or disable the Certificate after upload.

    **Note:** If you have selected the **Make default set** check box in the earlier page (Create new set or Add to existing set), then you need to associate the set to an operation mode. This will display certificate usages against operation modes. The encryption will be disabled for internal partners. SSL Client and Digital Signature will be disabled for external partners.
12. Click **Next** to go to the Configuration page of the wizard. In case you click **Finish** and there are some missing roots or intermediate CA certificates, you will be prompted to upload. If you click "Yes" in the prompt window, the first page of the wizard will open. Click **Cancel** if you want to upload at a later stage.
13. In the Configuration page of the wizard, enter the following values:

    **Note:** The Configuration page displays a list of certificate(set) usage against operation modes. The current set name is pre-populated for all, but you can reset it.
    - **From Partner** - This field will be pre-populated with the value of the internal partner.
    - **To Partner** - This drop-down is pre-populated with the list of all external partners. You can also select the value "All" to include all external partners.
    - **From Package** - From the drop-down, select the package Document Flow Definitions objects of the internal partner.
    - **To Package** - From the list, select the package Document Flow Definitions objects of the external partner.
14. Click **Add more connections** if you want to associate the set to other participant connections.
15. Click **Add Secondary Certificate** to add a secondary certificate to the current set.

16. Click **Finish** to upload the Certificate. In case there are some missing roots or intermediate CA certificates, you will be prompted to upload. If you click "Yes" in the prompt window, the first page of the wizard will open. Click **Cancel** in the prompt window if you want to upload at a later stage.

If you are uploading primary and secondary certificates for both SSL client authentication and digital signature and you are uploading the primary certificates as two separate entries, make sure that the corresponding secondary certificates are uploaded as two different entries.

### Obtaining a CA-signed certificate

#### About this task

If you are going to use a certificate signed by a CA, use the following procedure:

1. Start the iKeyman utility.
2. Use iKeyman to generate a certificate request and a key pair for the Receiver.
3. Submit a Certificate Signing Request (CSR) to a CA.
4. When you receive the signed certificate from the CA, use iKeyman to place the signed certificate into the key store.
5. Distribute the signing CA certificate to all partners.

## Installing an inbound digital signature verification certificate

#### About this task

The Document Manager uses the partner's signed certificate to verify the sender's signature when you receive documents. The partners send their self-signed signature certificates in X.509 DER format to you. You, in turn, install the partners' certificates through the Community Console under the respective partner's profile.

To install the certificate, use the following procedure.

1. Receive the partner's X.509 signature certificate in DER format.
2. Navigate to **Profile**> **External partner** > **Certificates** > **Load certificate**.
3. Click **Search** to find specific or sub-set of partners.
4. Click **Browse** next to **Certificate Location** to upload the Certificate.
5. Click **Next** to go to **Certificate Details** page of the wizard.
6. Associate this certificate to **Digital Signature Verification**.
7. Select **enabled** or **disabled** in the **Status** based on whether you want to enable or disable the Certificate after uploading.
8. Select the **Operation mode**. If you are a hub operator, you do not have the option to select the **Operation mode**.
9. Click **Finish** to save the changes and close the wizard.
10. If the certificate is signed by a CA and the CA root certificate and any other certificates that are part of the certificate chain are not already installed in the Hub Operator profile, install the certificates now. This is only applicable for Trust Store/Keystore.
    a. Click **Hub admin> Hub partner profile > Certificates** to display the Certificate List page.

       Make sure you are logged in to the Community Console as the Hub Operator, and install the certificate in your own profile.
    b. Click **Load Certificate**.
    c. Select **Root and Intermediate**.

d. Type a description of the certificate (which is required).
e. Change the status to **Enabled**.
f. Click **Browse** and navigate to the directory in which you have saved the certificate.
g. Select the certificate and click **Open**.
h. Click **Upload** and then click **Save**.

**Note:** You do not have to perform the previous step if the CA certificate is already installed.

11. Enable signing at the package (highest level), partner, or connection level (lowest level). Your setting can override other settings at the connection level. The connection summary will inform you if any required attribute is missing.

    For example, to alter the attributes of a partner connection, click **Account Admin > Connections** and then select the partners. Click **Attributes** and then edit the attribute (for example, **AS Signed**).

# Using certificates to enable SSL

The following sections describe how to create and install SSL certificates for use with WebSphere Partner Gateway. Also included is an overview of the SSL handshake process. If your community is not using SSL, neither you nor your partners need an inbound or outbound SSL certificate.

## SSL handshake

### About this task

Each SSL session begins with a handshake.

When a client (the partner or internal partner) initiates a message exchange, the following steps occur:

1. The client sends a client "hello" message that lists the cryptographic capabilities of the client (sorted in client preference order), such as the version of SSL, the cipher suites supported by the client, and the data compression methods supported by the client. The message also contains a 28-byte random number.
2. The server responds with a server "hello done" message that contains the cryptographic method (cipher suite) and the data compression method selected by the server, the session ID, and another random number.

   **Note:** The client and the server must support at least one common cipher suite, or else the handshake fails. The server generally chooses the strongest common cipher suite.
3. The server sends its digital certificate.

   Server authentication happens at this step.
4. The server sends a "digital certificate request" message. In the "digital certificate request" message, the server sends a list of the types of digital certificates supported and the distinguished names of acceptable certificate authorities.
5. The server sends a server "hello done" message and waits for a client response.

6. Upon receipt of the server "hello done" message, the client verifies the validity of the server's digital certificate and checks that the server's "hello" parameters are acceptable.
7. If the server requested a client digital certificate, the client sends a digital certificate, or if no suitable digital certificate is available, the client sends a "no digital certificate" alert. This alert is only a warning, but the server application can fail the session if client authentication is mandatory.
8. The client sends a "client key exchange" message. This message contains the premaster secret, a 46-byte random number used in the generation of the symmetric encryption keys and the message authentication code (MAC) keys, encrypted with the public key of the server.
9. If the client sent a digital certificate to the server, the client sends a "digital certificate verify" message signed with the client's private key. By verifying the signature of this message, the server can explicitly verify the ownership of the client digital certificate.

   **Note:** An additional process to verify the server digital certificate is not necessary. If the server does not have the private key that belongs to the digital certificate, it cannot decrypt the premaster secret and create the correct keys for the symmetric encryption algorithm, and the handshake fails.
10. The client uses a series of cryptographic operations to convert the premaster secret into a master secret, from which all key material required for encryption and message authentication is derived. Then the client sends a "change cipher spec" message to make the server switch to the newly negotiated cipher suite. The next message sent by the client (the "finished" message) is the first message encrypted with this cipher method and keys.
11. The server responds with a "change cipher spec" and a "finished" message of its own.

Client authentication requires steps 4 on page 255, 7, and 9.

The SSL handshake ends, and encrypted application data can be sent.

## Configuring inbound SSL certificates

This section describes how to configure server authentication and client authentication for inbound connection requests from partners.

An inbound request is when the partner is sending a document to WebSphere Partner Gateway. If your community is not using SSL, you do not need an inbound or outbound SSL certificate.

**Note:** For inbound FTPS WebSphere Partner Gateway uses an FTP Server that is provided by the customer, so any inbound SSL configuration is per that specific FTP Server product that the customer is using.

### Step 1: Obtain an SSL certificate

#### About this task

WebSphere Application Server uses the SSL certificate when it receives connection requests from partners through SSL. It is the certificate that the Receiver presents to identify the hub to the partner. This server certificate can be self-signed, or it can be signed by a CA. In most cases you will use a CA certificate to increase security. You might use a self-signed certificate in a test environment. Use iKeyman or the WebSphere Application Server administrative console to generate a

certificate and key pair. Refer to documentation available from IBM for more information about using iKeyman or the WebSphere Application Server administrative console.

After you generate the certificate and key pair, use the certificate for inbound SSL traffic for all partners. If you have multiple Receivers or Consoles, copy the resultant key store to each instance. If the certificate is generated using the WebSphere Application Server administrative console, the key and the certificate can be imported in another key store in another server using the WebSphere Application Server administrative console. If the certificate is self-signed, provide this certificate to the partners. To obtain this certificate, use iKeyman to extract the public certificate to a file.

**Generating a self-signed certificate:** If you are going to use self-signed server certificates, use the following procedure.

1. Start the iKeyman utility, which is located in /<*WAS_Installation_dir*>/bin. If this is your first time using iKeyman, delete the "dummy" certificate that resides in the key store.
2. Open the Receiver or Console key store using iKeyman, and use iKeyman to generate a self-signed certificate and a key pair for the Receiver or Console key store.
3. Use iKeyman to extract to a file the certificate that will contain your public key.

   Save the key store to a JKS, PKCS12, or JCEKS file.
4. Distribute the certificate to your partners. The preferred method for distribution is to send the certificate in a zipped file that is password-protected, by e-mail. Your partners must call you and request the password for the zipped file.
5. Using the WebSphere Application Server administrative console, set the new certificate in the SSL Configuration and in the settings for receiver and console. You can do this by selecting the alias of the new certificate in the key store in the Configuration for each node or server.

**Obtaining a CA-generated certificate:** If you are going to use a certificate signed by a CA, use the following procedure.

1. Start the iKeyman utility, which is located in the /<*WAS_Installation_dir*>/bin directory.
2. Use iKeyman to generate a certificate request and a key pair for the Receiver.
3. Submit a Certificate Signing Request (CSR) to a CA.
4. When you receive the signed certificate from the CA, use iKeyman to place the signed certificate into the key store.
5. Distribute the CA certificate to all partners if required.
6. Using the WebSphere Application Server administrative console, set the new certificate in the SSL Configuration and in the settings for receiver and console. You can do this by selecting the alias of the new certificate in the key store in the Configuration for each node or server.

   **Note:** The WebSphere Application Server administrative console can also be used to complete the previous steps.

## Step 2: Authenticate clients

### About this task

If you want to authenticate partners who send documents, perform the steps in this section.

**Installing the client certificate:**
**About this task**

For client authentication, use the following procedure:

1. Obtain your partner's certificate.
2. If the certificate is self-signed, install the certificate into the trust store using iKeyman or the WebSphere Application Server administrative console.
3. If the certificate is CA-issued, add the related CA certificates in the related trust store using iKeyman or the WebSphere Application Server administrative console.

**Note:** When you add more partners to your hub community, you can use iKeyman or the WebSphere Application Server administrative console to add their certificates to the trust store. If a partner leaves the community, you can use iKeyman or the WebSphere Application Server administrative console to remove the partner's certificates from the trust store.

**Setting up client authentication:**
**About this task**

After installing the certificate or certificates, configure WebSphere Application Server to use client authentication by running the utility script bcgClientAuth.jacl.

1. Navigate to the following directory: /<*ProductDir*>/bin
2. To turn on client authentication, call the script as follows:

   ```
   ./bcgwsadmin.sh -f /<ProductDir>/scripts/bcgClientAuth.jacl
      -conntype NONE set
   ```

   **Note:** To turn off client authentication, call the script as follows:

   ```
   ./bcgwsadmin.sh -f /<ProductDir>/receiver/scripts/bcgClientAuth.jacl
     -conntype NONE clear
   ```

You must restart the bcgreceiver server for these changes to take effect. Client Authentication can also be enabled using the WebSphere Application Server administrative console. A value of "Supported" means that server will ask for client certificate, but, if client certificate is not available, the SSL handshake may still be established. A value of "Required" means that client certificate must be sent. Otherwise, the SSL handshake will fail.

**Validating the client's certificate:**
**About this task**

There is an additional feature that can be used with SSL client authentication. This feature is enabled through the Community Console. For HTTPS, WebSphere Partner Gateway checks certificates against the Business IDs in the inbound documents. To use this feature, create the partner's profile, import the client certificate, and flag it as SSL.

1. Import the client certificate.
   a. Click **Account Admin > Profiles > Partner**, and search for the partner's profile.
   b. Click **Certificates**.
   c. Click **Load Certificate**.
   d. Click **Browse** and navigate to the directory in which you have saved the certificate.
   e. Select **SSL Client** as the type of certificate.

f. Type a description of the certificate (which is required).
g. Change the status to **Enabled**.
h. If you want to select a operation mode other than **Production** (the default), select it from the list.
i. Click **Finish**.

2. Update the client destination.
   a. Click **Account Admin > Profiles > Partner**, and search for the partner's profile.
   b. Click **Destinations**.
   c. Select the HTTPS destination you previously created. If you have not yet created the HTTPS destination, see "Setting up an HTTPS destination" on page 213.
   d. Click the **Edit** icon to edit the destination.
   e. Select **Yes** for **Validate SSL Client Certificate**.
   f. Click **Save**.

## Configuring separate keystore and truststore for receiver and console

By default, WebSphere Partner Gateway uses common keystore and truststore for the Receiver and Console. However, you can configure separate keystore and truststore for receiver and console in the distributed mode installation.

To configure the keystore and truststore, create and set a separate keystore and truststore for the Receiver and Console. Also, create separate SSL configurations. The SSL configurations can be set either at Cluster level or Server level. Setting SSL configuration at cluster level is easier since the configuration is then applicable to all the servers in that cluster, and you need not configure each server separately.

**Setting SSL configuration at the cluster level:** While setting the SSL configuration with new keystore and truststore at cluster level, there must not be any SSL configuration set at the server level. If there is a SSL configuration set at the server level, then the SSL configuration at the cluster level will not used; instead the one set for the server will be used.

Follow these steps to set the SSL configuration for bcgconsoleCluster:

1. Create a keystore for the Console cluster. The keystore must be created in the bcgconsole cluster scope by navigating to **Security > SSL certificate and key management > Key stores and certificates**.
2. Create a truststore for the Console cluster. The truststore must be created in the bcgconsole cluster scope by navigating to **Security > SSL certificate and key management > Key stores and certificates**.
3. Create an SSL configuration for console cluster at the Console cluster scope by navigating to **Security > SSL certificate and key management > SSL configurations**. Set the keystore and truststore that were created in the previous steps. Update the certificate aliases in the certificate aliases list by clicking **Get certificate aliases**, and select the required alias to be used for server authentication. Set the trust manager to **IbmPKIX**.
4. Set this SSL configuration in bcgconsoleCluster by overriding the inherited SSL configuration. Update the certificate aliases by clicking **Update the certificate aliases** and set the alias to be used for server authentication.
5. Restart bcgconsoleCluster.

Follow these steps to set the SSL configuration for bcgreceiverCluster:

1. Create a keystore for the Receiver cluster. The keystore must be created in the bcgreceiver cluster scope by navigating to **Security > SSL certificate and key management > Key stores and certificates**.
2. Create a truststore for the Receiver cluster. The truststore must be created in the bcgconsole cluster scope by navigating to **Security > SSL certificate and key management > Key stores and certificates**.
3. Create an SSL configuration for receiver cluster at the Receiver cluster scope by navigating to **Security > SSL certificate and key management > SSL configurations**, and set the keystore and truststore that were created in the previous steps. Get the certificate aliases by clicking **Get certificate aliases**, and select the required alias to be used for server authentication. Set the trust manager to **IbmPKIX**.
4. Set this SSL configuration in bcgreceiverCluster by overriding the inherited SSL configuration. Update the certificate aliases by clicking **Update the certificate aliases** and set the alias to be used for server authentication.
5. Restart the bcgreceiverCluster.

For more information on working with keystores, truststores, SSL configuration, and endpoint configurations, refer to the section *Securing applications and their environment of WebSphere Application Server Documentation.*

**Setting NodeDefaultTrustStore in NodeDefaultSSLSetting in distributed mode:** This setting must be done for simple distributed mode. But, this is also applicable for the fully distributed mode if common keystore and truststore are to be used for the Receiver and Console. If a node is federated in a cell, the signer certificates from the node are added to the CellDefaultTrustStore. By default, NodeDefaultSSLSetting refers to CellDefaultTrustStore as the truststore. For the WebSphere Partner Gateway Receiver and Console, using Signer certificates from other nodes might not be desirable. To use a dedicated truststore for the nodes in which WebSphere Partner Gateway is installed, NodeDefaultTrustStore can be set in NodeDefaultSSLSettings as the truststore.

The steps for making this change are as follows:

1. In the WebSphere Application Server administrative console, navigate to **Security > SSL certificate and key management > Manage endpoint security configurations > <node_name> > SSL configurations > NodeDefaultSSLSettings**.
2. In the field Trust store name, select **NodeDefaultTrustStore**.

   **Note:** Ensure that NodeDefaultTrustStore is configured for the truststore that you want to use; for example, bcgSecurityTrust.jks.
3. Click **Apply**.
4. On the following page of the Console, click **Save** to update the changes to the master configuration.
5. Restart the servers in that node.

**Note:** For the fully distributed mode, the above changes must be made for all nodes containing bcgreceiver and bcgconsole servers. For simple distributed mode, these changes must be made for all nodes containing bcgserver.

**Adding Signer certificates to trust.p12 if NodeDefaultTrustStore is set for node containing WebSphere Partner Gateway servers:** Currently, NodeDefaultTrustStore refers to trust.p12. If NodeDefaultTrustStore is set for the

node containing WebSphere Partner Gateway servers, bcgSecurityTrust.jks will not be used. Signer certificates from bcgSecurityTrust.jks needs to be added to trust.p12 as required.

## Configuring outbound SSL certificates

An outbound request is when WebSphere Partner Gateway is sending a document to a partner. If your community is not using SSL, you do not need an inbound or outbound SSL certificate.

### Step 1: Authenticate the server

#### About this task

When SSL is being used to send outbound documents to your partners, WebSphere Partner Gateway requests a server-side certificate from the partners. The same CA certificate can be used for multiple partners. The certificate must be in X.509 DER format.

**Note:** You can convert the format with the iKeyman utility. Follow these steps to use iKeyman to convert the format:

1. Start iKeyman.
2. Create a new blank key store or open an existing key store.
3. In the Key Database Content, select **Signer Certificates**.
4. Add the ARM certificate using the **Add** option.
5. Extract the same certificate as a Binary DER data using the **Extract** option.
6. Close iKeyman.

Install the partner's self-signed certificate into the Hub Operator profile. If the certificate was signed by a CA and the CA root certificate and any other certificates that are part of the certificate chain are not already installed in the Hub Operator profile, install the certificates in the Hub Operator profile.

1. Click **Account Admin > Profiles > Certificates** to display the Certificate List page.

   Make sure you are logged in to the Community Console as the Hub Operator or Internal Partner.
2. Click **Load PKCS12.**.

   **Note:** The PKCS12 file being uploaded should contain only one private key and the associated certificate. You can also upload the certificate and the PKCS#8-formatted private key separately.
3. Select **SSL Client** as the type of certificate.
4. Type a description of the certificate (which is required).
5. Change the status to **Enabled**.
6. Click **Browse** and navigate to the directory in which you have saved the certificate.
7. Select the certificate and click **Open**.
8. Enter the password.
9. If you want to select a operation mode other than **Production** (the default), select it from the list.
10. If you have two SSL certificates, indicate whether this is the primary or secondary certificate by selecting **Primary** or **Secondary** from the **Certificate Usage** list.

11. Click **Upload** and then click **Save**.

**Note:** You do not have to perform the previous steps if the CA certificate is already installed.

## Step 2: Authenticate clients
### About this task

If SSL client authentication is required, the partner will, in turn, request a certificate from the hub. Use the Community Console to import your certificate into WebSphere Partner Gateway. You can generate the certificate using iKeyman. If the certificate is a self-signed certificate, it must be provided to the partner. If it is a CA-signed certificate, the CA root certificate must be given to the partners, so that they can add it to their trusted certificates.

You can have more than one SSL certificate. One is the primary certificate, which is the one used by default. The other is a secondary certificate, which is used if the primary certificate expires.

**Using a self-signed certificate:**
**About this task**

If you are going to use a self-signed certificate, use the following procedure.

1. Start the iKeyman utility.
2. Use iKeyman to generate a self-signed certificate and a key pair.
3. Use iKeyman to extract to a file the certificate that will contain your public key.
4. Distribute the certificate to your partners. The preferred method for distribution is to send the certificate in a zipped file that is password-protected, by e-mail. Your partners must call you and request the password for the zipped file.
5. Use iKeyman to export the self-signed certificate and private key pair in the form of a PKCS12 file.
6. Install the self-signed certificate and key through the Community Console.
   a. Click **Account Admin > Profiles > Certificates** to display the Certificate List page.

      Make sure you are logged in to the Community Console as the Hub Operator.
   b. Click **Load PKCS12**.

      **Note:** The PKCS12 file being uploaded should contain only one private key and the associated certificate. You can also upload the certificate and the PKCS#8-formatted private key separately.
   c. Select **SSL Client** as the type of certificate.
   d. Type a description of the certificate (which is required).
   e. Change the status to **Enabled**.
   f. Click **Browse** and navigate to the directory in which you have saved the certificate.
   g. Select the certificate and click **Open**.
   h. Enter the password.
   i. If you want to select a operation mode other than **Production** (the default), select it from the list.

j. If you have two SSL certificates, indicate whether this is the primary or secondary certificate by selecting **Primary** or **Secondary** from the **Certificate Usage** list.

k. Click **Upload** and then click **Save**.

If you are uploading primary and secondary certificates for both SSL client authentication and digital signature and you are uploading the primary certificates as two separate entries, make sure that the corresponding secondary certificates are uploaded as two different entries.

**Using a CA-signed certificate:**
**About this task**

If you are going to use a certificate signed by a CA, use the following procedure:

1. Use iKeyman to generate a certificate request and a key pair for the Receiver.
2. Submit a Certificate Signing Request (CSR) to a CA.
3. When you receive the signed certificate from the CA, use iKeyman to place the signed certificate into the key store.
4. Distribute the signing CA certificate to all partners.

## Adding a Certificate Revocation List (CRL)

WebSphere Partner Gateway includes a Certificate Revocation List (CRL) feature. The CRL, issued by a Certificate Authority (CA), identifies partners who have revoked certificates before their scheduled expiration date. Partners with revoked certificates will be denied access to WebSphere Partner Gateway.

Each revoked certificate is identified in a CRL by its certificate serial number. The Document Manager scans the CRL every 60 seconds and refuses a certificate if it is contained within the CRL list. However, you can configure the time interval at which the CRL directory is scanned. The time interval is specified for the configuration property bcg.rosettanet.encrypt.CertDbRefreshInterval.

By default, CRLs are stored in the following location: */<shared_data_directory>/security/crl*. WebSphere Partner Gateway uses the setting bcg.CRLDir in Console > System Administration > DocMgr Administration > Security to identify the location of the CRL directory.

Place the CRLs in the CRL directory.

## Configuring CRLDP

### About this task

Configure the CRLDP by changing the Java Virtual Machine settings, that is, set the value of -Dcom.ibm.security.enableCRLDP = True.

In fully distributed mode, this setting has to be done for bcgdocmgr, bcgreceiver, and bcgconsole. In case of simple distributed mode and simple mode, carry out this setting for bcgserver.

The steps are as follows:

1. Log in to the WebSphere Application Server Admin console.
2. Go to **Servers > Application servers** and select **Server**.
3. Set the property using the following process:

a. Select the server (bcgdocmgr, bcgreceiver, or bcgconsole).
b. In the **Configuration** page, expand **Java and Process Management** in the **Server infrastructure** section of the page and select **Process Definition.**
c. In the **Process definition configuration** page, select **Java Virtual Machine** in the **Additional Properties** section.
d. Append the following to the existing value (if existing) in the Generic JVM Arguments field: -Dcom.ibm.security.enableCRLDP=true.

4. Click **Apply** and then Save to complete this configuration.
5. Restart the server.
6. Set this property in all the servers in the cluster.

## Configuring inbound SSL for the Community Console and Receiver component

The WebSphere Partner Gateway key stores are preconfigured in WebSphere Application Server. This section applies only if you are using different key stores.

To configure SSL for the Community Console and Receiver component in WebSphere Partner Gateway, use the following procedure.

1. Obtain the following information:
   - The full path names of the key file and the trust file; for example for the Receiver: <*ProductDir*>/common/security/keystore/bcgSecurity.jks and <*ProductDir*>/common/security/keystore/bcgSecurityTrust.jks

     You must enter these names correctly. In the UNIX environment, these names are case-sensitive.
   - The new passwords for each file.
   - The format of each file. This must be chosen from one of the values JKS, JCEKS, or PKCS12. Enter this value in uppercase exactly as shown.
   - The path to the script file named bcgssl.jacl.
2. Open a Community Console window and change to /<*ProductDir*>/bin. The server does not need to be running to change the passwords.
3. Enter the following command, substituting the values that are enclosed in <>. All values must be entered.

```
./bcgwsadmin.sh -f /<ProductDir>/
scripts/bcgssl.jacl -conntype NONE install
<keyFile_pathname>
<keyFile_password> <keyFile_format> <trustFile_pathname>
<trustFile_password> <trustFile_format>
```

4. Start the server. If the server fails to start, it might be because of an error when running bcgssl.jacl. If you make a mistake, you can rerun the script to correct it.
5. If you used bcgClientAuth.jacl to set the clientAuthentication SSL property, reset it after using bcgssl.jacl. This is because bcgssl.jacl overwrites any values that might have been set for client authentication with the value false.

**Note:**

1. Repeat these steps for the Console, substituting **console** for **receiver** in the path name.
2. The configuration for SSL, key store and trust store can also be done using the WebSphere Application Server administrative console.

By default, WebSphere Partner Gateway supports one keystore and truststore for both receiver and the console. However, you can use separate keystore and

truststore for receiver and console in Full Distributed mode. To use separate keystore and truststore for receiver and console, do the following configuration using WAS Admin Console for the Receiver:

1. Create a keystore for the receiver keystore. Refer to section Creating a keystore configuration in the WAS documentation.
2. Create a truststore for the receiver truststore. Refer to section <*Creating a keystore configuration*in the WAS documentation <*Securing applications and their environment.*
3. Create a SSL configuration for the receiver and set the above keystore and truststore in that configuration. Select the required alias to be used for server authentication in the keystore. Set the trust manager to **IbmPKIX**. Refer to the section *Creating a Secure Socket Layer configuration* in the WAS documentation *Securing applications and their environment.*
4. Set this SSL configuration in each bcgreceiver server by overriding the inherited SSL configuration. Set the alias to be used for the server authentication.
5. Restart each bcgreceiver server.

The steps are similar for console configuration. Refer to appropriate sections in the WAS documentation *Securing applications and their environment.*

1. Create a keystore for the console keystore.
2. Create a truststore for the console truststore.
3. Create a SSL configuration for console and set the above keystore and truststore in that configuration. Select the required alias to be used for server authentication in the keystore. Set the trust manager to **IbmPKIX.**
4. Set this SSL configuration in each bcgconsole server by overriding the inherited SSL configuration. Set the alias to be used for server authentication.
5. Restart each bcgconsole server.

For more information on working with keystores, truststores, SSL configuration, and endpoint configurations, refer WAS Documentation, *Securing applications and their environment*.

**Note:** Currently, NodeDefaultTrustStore refers to trust.p12. If NodeDefaultTrustStore is set for bcg node, then bcgSecurityTrust.jks will not be used. You need to add Signer certificates from bcgSecurityTrust.jks to trust.p12 as required.

# Uploading certificates using wizard

## About this task

As a Hub Operator you can upload certificates for internal or external partners:

- Upload private key and certificates for internal partners.
- Upload public certificates for external partners.
- Upload root and intermediate CA certificate.

  **Important:** This functionality is available for X.509 certificates only.
- Upload a certificate chain from a trust store.

  **Important:** This functionality is available for X.509 certificates only.

Certificate upload wizard is provided to upload certificates. Using the wizard you can set the usage of the certificate (Sign / verify / Encrypt / decrypt /SSL client/FTPS server/SFTP server), associate it to one or more operation modes, add

to a set (either an existing one or a new one), select the certificate to be the default for all the participant connections or select a specific connection where this certificate set will be used. The option of associating the certificate to the connection will not appear if the certificate is not associated to a set. While uploading the certificate, make sure that the certificate has not expired.

For OpenPGP, a public key packet file can also be used to upload the public key and certificate of an external partner. The external partner can export the key from the keyring and store it in a file and send it to hub operator. The hub operator can upload the certificate received from the external partner. The public key file can be in binary or ASCII armor format.

Steps to upload certificates for partners (internal or external) using the wizard:

1. Select the partner and click **Account admin** > **Profile** > **Certificates**.
2. Click **Load Certificate**.
3. In the **Select Partner, File Location, Password** page of the wizard, enter the following values:
   - **Which partner does this certificate(s) belongs to**: Select the partner to associate the newly uploaded certificate. Click **Search** to find a specific partner or subset of a partners. If the partner is a Hub Operator or Internal Partner, enter the certificate location, private key location, and password (OR) Provide the truststore or keystore with password. For External Partner, enter the certificate location (OR) provide the trust store location containing the certificate chain.

     **Note:** If you click **Load Certificate** without selecting a partner profile, then the **Which partner does this certificate belong to** field is not displayed. The certificate is uploaded for the selected partner profile automatically.
   - **Is this a root and intermediate certificate**: Select this check box if the certificate is a root and intermediate certificate.

     **Note:** The Root and Intermediate certificate type applies only to the hub administrator profile, so the Root and Intermediate Certificate check box is visible only when the selected partner is the hub administrator. In addition, for hub administrator profiles, the Root and Intermediate Certificate check box is available only if you select Load Certificate.
   - **Certificate Location**: Click **Browse** to select the location of the certificate (public/private).
   - **Private Key:** Click **Browse** to select the Private Key of the certificate.

     **Note:** This is applicable only for an internal partner.
   - **Password:** If the certificate has a password, enter the value.
   - **Trust Store (or) Keystore Location or Keyring Location:** Click **Browse** to select the trust store (or) Keystore Location or keyring location. Trust store is a file that contains a collection of trusted CA and root certificates. Key store is a collection of private keys along with trusted root and CA certificates. Keyring is a collection of certificates in OpenPGP format. Click **Browse** to select the file from the File dialog path of the key ring/keystore/truststore file, or type the path in the text field. When you upload a certificate for an internal partner of type OpenPGP keyring, provide the path of the secret keyring file. For external partner, provide the path of the public keyring file.

- **Password:** If the Trust store (or) Keystore Location has a password, enter the value. In case of Keyring, password is not required.
- **Type:** Select the type of Trust store (or) Keystore or keyring location. The available values in the list are: JKS, JCEKS, PKCS12, and OpenPGP.

4. Click **Next**.
5. The **End entity and CA certificates** page of the wizard will open when you load certificates via a trust store that has more than one certificate. The list of certificates available in the trust store is displayed. The **Select the OpenPGP Keys/Certifications to be uploaded** is displayed when you select a Keyring of type OpenPGP in the **Select Partner, File Location, Password** page of the wizard.
   - Select a certificate in **End entity certificate to upload** page of the wizard. If the Key store has multiple private keys, then along with the private key you must enter the password for the key if different. In the **End entity certificate and CA Certificate** page of the wizard, enter the following values:
     – **The keystore contains more than one End Entity certificate. Select the certificate to be uploaded**- it has a list of all the End Entity certificates. Select the certificate to upload.
     – **Password** - if the keystore has a password, select this check box and enter the password in the text box.
     – **Select the List of Root and Intermediate CA certificates to be uploaded** - from the list box, select the Root and Intermediate CA certificates to upload.
   - In the **Select the OpenPGP Keys/Certifications to be uploaded** page of the wizard, the certificates associated with the selected keyring are populated in the list.

     **Note:** You can click **View details** to view the details of the selected certificate. In case of certificate, whenever the key ID and issuer ID are the same, the certificate is a self certificate.
     – Select a top-level key from the list.
     – Enter the password for the top-level key, if you want to upload the top-level key.

       **Note:** If there are any sub keys for the top level key, then all the sub keys are displayed in the **Select the sub key to be uploaded** list.

       **Important:** This is not applicable while loading public certificates for encryption.
     – Select the sub key, if required.
     – Enter the password of the sub key.

     **Important:** This is not applicable while loading public certificates for encryption.

     **Remember:** When you upload the certificate for an external partner, password for top-level and sub-key are not required.
6. Click **Next** to go to **Certificate Details** page of the wizard.
7. In the **Certificate Details** page of the wizard, enter the following details of the certificate:

- **Leaf Certificate Name** - the name of the Leaf Certificate. The field name depends on whether the certificate is a Leaf certificate, Root CA certificate or an intermediate CA certificate.
- **Description** - the description of the Leaf Certificate.
- **Is this certificate for FTP Server Authentication** - select this check box if the uploaded certificate is for FTP Server authentication.
- **Is this certificate for SFTP Server Authentication** - select this check box if the uploaded certificate is for SFTP Server authentication.

  **Important:** The FTP Server and SFTP Server authentication is not applicable for OpenPGP certificates.
- **Certificate Type** - associate this certificate to a certificate type. The different types supported are Digital Signature, Digital Signature Verification, Encryption, Decryption, SSL Server, and SSL Client.

  **Remember:**
  - Encryption option is for an external partner and Decryption option is for an internal partner.
  - SSL Client is not supported for OpenPGP certificate type.
- **Certificate Usage** - associate an usage for the certificate. The values are Primary and Secondary.

  **Important:** This is not applicable for decryption, signature verification, and SSL server certificate.
- **Operation Mode** - select an operation mode for encryption, signing and SSL client certificates.

  **Important:** Operation mode is not applicable for encryption and signature verification.
- **Status** - select enabled or disabled based on whether you want to enable or disable a certificate after upload. The **Next** button is enabled only if the certificate is enabled.
- **Set Management** - you can either associate a certificate to an existing set or create a new set. If the certificate is a secondary certificate, it can only be associated to an existing set. You can associate the certificate to any set for an internal partner with type encrypt or for a external partner with type SSL (Incoming client auth) or Signing (Verify).

  **Note:** In case of uploading a OpenPGP certificate for an internal partner, Set management is not applicable. In case of encryption for external partner and signing or SSL client certificate for internal partner, select either **Add New Set** or **Update Existing Set**. This is applicable only if you decide to use sets. Otherwise, click **Finish**.

8. Click **Next** to go to Set page of the wizard. If the certificate is primary, you do not have to create sets and associate the certificate to a set and participant connection. If you have selected **Create new set** check box, then **Create New Set** page of the wizard will open. Otherwise, the **Add to Existing** page of the wizard will open. If the file contains a private key of the internal partner or the public certificate of the external partner used for SSL / Digital Signature, then you can click **Finish**.

   **Important:** Secondary to primary transition is not supported for OpenPGP certificates.

9. In the **Create New Set** page of the wizard, enter the details of the new set. For Primary certificates, you do not have to create sets and associate a certificate to it. Enter the following values:
   - **Set Name** - The name of the Set.
   - **Description** - The description of the Set.
   - **Status** - Select enabled or disabled. If it is disabled the **Next** button will not be enabled.
   - **Make default settings** - Select this check box if you want this set to be the default.
10. In the **Add to Existing Set** page of the wizard, select set(s) to add the certificate. Enter the following values:
   - **Select from the list of Sets available for the selected Certificate type** - From the list, select set(s) to add the certificate.
   - **Make default settings** - Select this check box if you want this set to be the default.
11. From the **Create New Set** or **Add to Existing Set**, click **Next** to go to the **Default Settings** page of the wizard. The **Next** button will be enabled only if the status of the set is enabled.
12. Select **enabled** or **disabled** in the **Status** based on whether you want to enable or disable the Certificate after upload.

    **Note:** If you have selected the **Make default set** check box in the earlier page (Create new set or Add to existing set), then you need to associate the set to an operation mode. This will display certificate usages against operation modes. The encryption will be disabled for internal partners. SSL Client and Digital Signature will be disabled for external partners.
13. Click **Next** to go to the Configuration page of the wizard. In case you click **Finish** and there are some missing roots or intermediate CA certificates, you will be prompted to upload. If you click "Yes" in the prompt window, the first page of the wizard will open. Click **Cancel** if you want to upload at a later stage.
14. In the Configuration page of the wizard, enter the following values:

    **Note:** The Configuration page displays a list of certificate (set) usage against operation modes. The current set name is pre-populated for all, but you can reset it.
   - **From Partner** - This field will be pre-populated with the value of the internal partner.
   - **To Partner** - This list is pre-populated with the list of all external partners. You can also select the value "All" to include all external partners.
   - **From Package** - From the list, select the package Document Flow Definitions objects of the internal partner.
   - **To Package** - From the list, select the package Document Flow Definitions objects of the external partner.
15. Click **Add more connections** if you want to associate the set to other participant connections.
16. Click **Add Secondary Certificate** to add a secondary certificate to the current set.
17. Click **Finish** to upload the Certificate. In case there are some missing roots or intermediate CA certificates, you will be prompted to upload. If you click

"Yes" in the prompt window, the first page of the wizard will open. Click **Cancel** in the prompt window if you want to upload at a later stage.

**Note:** In case of OpenPGP, if a certificate upload failure occurs in spite of loading the correct certificate, then restart the server.

## Creating Certificate sets

### About this task

Certificate set is introduced for the following security functions:

- SSL Client Authentication of outbound messages from internal partner to external partner.
- Adding digital signature to outbound messages from internal partner to external partner.
- Encrypting outbound messages from internal partner to external partner.
- Sets are not used for inbound scenarios, such as verifying external partner's SSL client authentication certificate in WebSphere Partner Gateway trust store, verifying digital signature of external partner, and decrypting encrypted messages meant for internal partner.

To create a new Certificate set, follow this procedure:

1. In the Console, navigate to **Profile** > **Partner** > **Certificate List** > **Certificates Sets List** > **Create Set**.
2. Click **Certificate > Certificates Sets > Create Set**.
3. Enter the **Set Name** and **Description** for the new certificate set.
4. Set the **Certificate Type**.
5. Select **Enabled** or **Disabled** check box to enable or disable **Certificate Set**.
6. Click **Load Certificate**

   **Note:** The **Primary Certificate** and **Secondary Certificate** drop-down is populated based on the **Certificate Type** selected. If there are certificates already created and not associated to any set, then you can add those certificates to the set currently being created. If the certificates list is empty, then there will be a empty drop-down.
7. Select **Primary Certificate** and **Secondary Certificate** from the drop-down.
8. Click **Save**.

## Deleting Certificate set

### About this task

1. In the Console, navigate to **Profile** > **Partner** > **certificate set list**. This view lists out all the certificates created for the partner.
2. Click **Delete** icon. Before delete operation, makes sure that you have modified all the references to this set in the connection.
3. If the set is used by one or more connections, a warning message appears. To check where all a particular Certificate is used, see “Certificate Whereused” on page 271.
4. In the warning message window, click **OK** to delete or click **Cancel** to abort the deletion of certificate set.

## Certificate Whereused

In the Console, navigate to **Profile** > **{Partner}** > **Certificate List** > **Certificates Sets List** > **Whereused** . The resultant view displays the following details:

- From partner
- To Partner
- From package
- To Package
- SSL Client
- Digital Signature
- Digital Signature Verification
- Encrypt
- Decrypt
- Validity.

**Note:** The Certificate may be invalid because of the following reasons: Invalid if there is no primary certificate, Primary certificate is disabled, Set is disabled, Primary expired and there is no Secondary, and both Primary and Secondary expired.

## Setting up SSL for FTP Scripting receiver/destination

For FTP Scripting Receiver, the SSL client authentication certificate is loaded to the Hub-operator profile. Even if the certificates are loaded for internal partner, it will not override the global settings.

## Providing default certificate set for all internal partners

As WebSphere Partner Gateway supports multiple internal partners, each internal partner has to upload private keys. If an organization that wants to share a certificate with its organization units, you have to upload the certificate for each internal partner. To simplify this, you can provide a default option so that a particular certificate is used for all internal partners.

In the Console, navigate to **Certificates** > **Upload Certificates**. Upload the Certificates and provide details of certificate type, usage and operation mode. When you save the specified information, the certificate/keys are loaded at the hub-operator level. During runtime the default provided at the hub-operator level is used in the absence of any certificate.

## Certificate summary

Table 32 summarizes the way certificates are used in WebSphere Partner Gateway. Certificate locations are shown in parenthesis "( )".

*Table 32. Certificate summary information*

| Message delivery method (See note 1) | Hub operator certificate | Obtain certificate and CA from partner | CA (See note 2) | Give certificate to partner (See note 3) | Comments |
|---|---|---|---|---|---|
| Inbound SSL | Install on WebSphere Application server-side SSL. (Place in the WebSphere Application Server key store.) | Partner's self-signed certificate. | Only needed if client authentication is used. (Place the CA or self-signed certificate in the WebSphere Application Server trust store.) | Hub operator certificate if self-signed or the CA root certificate, if required, if it is CA-authenticated. | |
| Outbound SSL | If client authentication is being used. (WebSphere Partner Gateway) | Partner server-side certificate or CA root certificate if it is CA-authenticated. | WebSphere Partner Gateway | Hub Operator certificate if self-signed or CA certificate if signed by a third party. | |
| Inbound Decryption | Private key (WebSphere Partner Gateway) | N/A | If the certificate is CA-signed, CA certificates need to be uploaded as Root/ Intermediate certificates. | Hub Operator certificate | For decrypting the message |
| Inbound Digital Signature Verification | N/A | Certificate for validating the certificate used for the digital signature. (WebSphere Partner Gateway) | WebSphere Partner Gateway | NA | For verification and nonrepudiation |
| Outbound Encryption | N/A | Use the certificate obtained from the partner. (Certificate is installed in the partner's profile) | CA certificate chain for client certificate if not self-signed | N/A | For encryption of outbound messages |
| Outbound Signature | Private key and certificate (WebSphere Partner Gateway) | N/A | CA certificate chain. | Optional, depending on partner; give WebSphere Partner Gateway certificate | |

*Table 32. Certificate summary information (continued)*

| Message delivery method (See note 1) | Hub operator certificate | Obtain certificate and CA from partner | CA (See note 2) | Give certificate to partner (See note 3) | Comments |
|---|---|---|---|---|---|
| Certificate to business ID validation | N/A | Load in partner profile | | | Validates that this certificate is for this business ID when the SSL Client check is done |

**Notes:**

1. An inbound message is one coming into WebSphere Partner Gateway from a partner. An outbound message is one going out of WebSphere Partner Gateway to a partner.
2. If the certificate is CA-issued, the issuing CA certificate must be obtained and stored. This applies to either the Hub Operator certificate or the partner's certificate.
3. If a private key is involved, this certificate corresponds to the private key.

## Using PEM formatted certificate and key with WebSphere Partner Gateway

This section provides information on the usage of PEM encoded keys and certificates.

### Using PEM formatted private key

If you have a private key in PEM format and want to upload it into WebSphere Partner Gateway, then upload is not possible unless the private key is converted to PKCS#8 format.

This can be done using OpenSSL tool.

Use this command to convert a PEM formatted key to PKCS#8 format:

```
openssl pkcs8 -topk8 -in usr.key -out usr.p8 -outform DER
```

This command works for a key created using OpenSSL.

OpenSSL is available with Linux distributions and can also be downloaded from the web site http://www.openssl.org.

### Using PEM formatted certificate

The certificate can be uploaded in WebSphere Partner Gateway in PEM format. It works for a PEM formatted certificate generated using OpenSSL.

### PKCS#7 encoded certificate with WebSphere Partner Gateway

#### About this task

On Windows, if you have certificates encoded in PKCS#7 format (.p7b file), then perform the following steps to extract the certificates from the .p7b file:

1. Double-click the .p7b file.

2. In the navigation panel, expand the folder tree and click **Certificates**. The list of certificates contained in the file is displayed on the right side.
3. To copy a certificate to the file system, double-click the certificate. The details of the certificate are displayed.
4. In the certificate details, click **Details** tab.
5. Click **Copy to file** to copy the file to file system.
6. Export the certificate as a DER encoded file.

## Loading SFTP Keys

Steps to load SFTP Keys.

### About this task

To load SFTP Keys, perform the following steps:

1. Navigate to **Account Admin** > **Profiles** > **Certificates**.
2. Click **Load SFTP Keys**.
3. In the **Load SFTP Keys** page, click **Browse** and select the Key file from your local. The uploaded file is used for key based authentication. The **Data contained** icon indicates that a Key is already uploaded

## FIPS compliance

WebSphere Partner Gateway is compliant to FIPS (Federal Information processing Standard) FIPS 140-2 standard. **IBMJCEFIPS** is the FIPS-compliant JCE provider. **IBMJSSE2 JSSE** provider uses **IBM JCE** and does not contain code for cryptography, so it is not necessary to be certified for FIPS compliance. Though **IBMJSSEFIPS JSSE** provider is FIPS-compliant, use **IBMJSSE2** provider in WebSphere Partner Gateway. **IBMJSSE2** is the latest provider and it supports more algorithms and has improved serviceability. The product can be run in either FIPS mode or non-FIPS mode. If FIPS mode is configured and non-FIPS approved algorithm is used, an error event is generated and document transaction is stopped. PKCS#12 algorithm is not FIPS approved, so PKCS#12 files cannot be uploaded in FIPS mode. You must be an administrator to configure WebSphere Partner Gateway to run in FIPS or default mode. For FIPS mode, PKCS#12 can be uploaded to the console of WebSphere Partner Gateway in JCEKS or JKS format using iKeyman.

FIPS mode supports JKS, JCEKS, and OpenPGP keystores, but does not support PKCS#12 keystores. From the Console, you can upload a certificate and key in JKS or JCEKS or OpenPGP format. In **Keystore upload** screen, select the format from the **Keystore format** list. The values available in **Keystore format** list are: PKCS#12, JKS, JCEKS, and OpenPGP.

### Configuring WebSphere Partner Gateway to run in FIPS mode

#### About this task

To configure the WebSphere Partner Gateway to run in FIPS mode, use the following procedure:

1. Set the FIPS providers in **java.security** file.
2. Set the **bcg.FIPSMode** system property to "true" in the Console of WebSphere Partner Gateway.

3. Set the IBMJCEFIPS provider before the IBMJCE provider in the java.security file. The java.security file is in <WAS Installation>/java/jre/lib/security directory.
4. Set the FIPS enabled Socket factory classes for JSSE socket factory and server socket factory.
5. Restart all the servers.

   **Note:** An informational event is generated to indicate that the product is running in FIPS mode.

## Configuring WebSphere Partner Gateway to run in default mode

### About this task

To configure the WebSphere Partner Gateway to run in default mode, use the following procedure:

1. In the Console of WebSphere Partner Gateway, set the **bcg.FIPSMode** system property to "False".
2. Reset the settings for JSSE socket factory, server socket factory and the providers in **java.security** file as mentioned below:
   a. Remove the **com.ibm.jsse2.JSSEFIPS=true** system property from the Generic JVM Properties for each server.
   b. Reset the values of the following properties to their original values:
      - ssl.SocketFactory.provider
      - ssl.SocketFactory.provider
   c. For every WAS installation, comment IBMJCEFIPS provider and renumber the providers, starting from 1, in the **java.security** file.
3. Restart the servers.

   **Note:** An informational event is generated to indicate the mode. In default mode, all supported algorithms can be used including non-FIPS approved algorithms.

## Configuring IBM JSSE providers for FIPS mode

### About this task

To configure IBM JSSE providers for FIPS mode, use the following procedure:

1. Set the **com.ibm.jsse2.JSSEFIPS** system property to "True". This is done by setting JVM system properties for the application server using the WAS Admin console. Navigate to the page <Server>/Java and Process Management/Process Definition/Java Virtual Machine and specify the property **-Dcom.ibm.jsse2.JSSEFIPS=true**. This setting has to be done for each server.
2. Set the following security properties for the IBMJSSE2 Provider to handle all JSSE requests:
   - ssl.SocketFactory.provider = com.ibm.jsse2.SSLSocketFactoryImpl
   - ssl.ServerSocketFactory.provider = com.ibm.jsse2.SSLServerSocketFactoryImpl
3. Add IBMJCEFIPS provider, com.ibm.crypto.fips.provider.IBMJCEFIPS, to the provider list before the IBMJCE provider. Do not remove the IBMJCE provider, as it is required for KeyStore support.

   **Note:** Only the TLS protocol is supported when IBMJSSE2 is in FIPS mode.

## Algorithms supported in FIPS and non-FIPS mode

Following algorithms are supported in FIPS:

- Diffie-Hellman
- RSA, DSA
- SHA-1, SHA-384, SHA-224, SHA-512
- AES, DES, TDES (Triple DES)
- FIPS 186-2 – Algorithm for Pseudo Random Number generation (PRNG)
- Transport layer security: TLSv1
- Keystore format: JKS, JCEKS

Following algorithms are supported in WebSphere Partner Gateway:

- Asymmetric Cryptography: RSA, DSA
- Hash Function: SHA-1, MD5, SHA-384, SHA-224, SHA-512, RIPEMD/160.
- Symmetric Cryptography: AES, DES, 3DES, RC2 (All with CBC mode), CAST5, Blowfish, Twofish.
- PRNG: IBMSecureRandom
- Signature algorithm: dsa-sha1, rsa-sha1
- Transport layer security: SSLv3, TLSv1
- Keystore format: PKCS#12, JKS, JCEKS, OpenPGP
- Symmetric key algorithms: AES and TripleDES with modification detection.

  **Restriction:** You can use TripleDES and AES algorithms only when both Modification Detection and FIPS mode are set.

The following algorithms are not supported in FIPS but are supported in WebSphere Partner Gateway:

- Hash function: MD5. RIPEMD160
- Symmetric cryptography: RC2, CAST5, Blowfish, Twofish
- IBMSecureRandom PRNG provider (all cases of WebSphere Partner Gateway).
- Transport layer security: SSLv3
- Keystore format: PKCS#12

# Chapter 14. Managing alerts

WebSphere Partner Gateway's alerts are used to notify key personnel of unusual fluctuations in the volume of transmissions you receive, or when business document processing errors occur.

A companion option in the Viewer module, Event Viewer, helps you further identify, it, and resolve processing errors.

## Overview of alerts

An alert consists of a text-based e-mail message sent to subscribed contacts or a distribution list of key personnel. Alerts are based on the occurrence of a system event (event-based alert) or expected document flow volume (volume-based alert).

- Use a **volume-based alert** to receive notification of an increase or decrease in the volume of transmissions.

  For example, if you are an external partner, you can create a volume-based alert that notifies you if you do not receive any transmissions from the internal partner on any business day (set Volume to Zero Volume, set frequency to Daily, and select Mon through Fri in the Days of Week option). This alert can highlight internal partner network transmission difficulties.

  If you are an external partner, you can also create a volume-based alert that warns you when the number of transmissions from the internal partner exceeds the normal rate. For example, if you normally receive approximately 1000 transmissions a day, you can set the Expected Volume at 1000 and the Percent Deviation at 25%. The alert will notify you when you receive more than 1250 transmissions a day (it will also notify you when the volume of transmissions falls below 750). This alert can identify increased demand on the part of the internal partner, which might, over time, require you to add more servers to your environment. For more information on volume-based alerts, see " Creating a volume-based alert" on page 280.

  **Note:**

  1. Volume-based alerts monitor volume with respect to the document type that you select when you create the alert. WebSphere Partner Gateway only looks at documents that contain the document type selected in your alert, and generates alerts only when all of the alert criteria are met.
  2. The external partner can only create a volume-based alert on the volume of documents sent to the internal partner. For the external partner to set up a volume-based alert on the volume of incoming documents from the internal partner, the external partner would request the hub administrator to set up a volume-based alert on the external partner's behalf, specifying the external partner as the alert owner. An internal partner can also create volume-based alerts to send to external partners.

- Use an **event-based alert** to receive notification when errors in document processing occur. For example, you might want to create an alert that notifies you if your documents fail processing due to validation errors or because duplicate documents were received. You can also create alerts that let you know when a certificate is about to expire.

  You will use WebSphere Partner Gateway predefined event codes to create event-based alerts. There are five event types: Debug, Information, Warning,

Error, Critical. Within each event type, there are many events. You can view and select predefined events on the Alert: Events page. For example, 240601 AS Retry Failure, or 108001 Not a Certificate. For more information on event-based alerts, see “ Creating an event-based alert” on page 282.

**Tip:**

- Use a volume-based alert to receive notification if expected external partner or internal partner transmission volume falls below operating limits. This alert can highlight external partner or internal partner network transmission difficulties.
- Use an event-based alert to receive notification of errors in document processing. For example, you can create an event-based alert that notifies you if your documents have failed processing due to validation errors.

**Note:** To send alerts, you must configure an e-mail server for the alerts. Alerts are configured in the Alert Engine Attributes page found by clicking **System Administration** > **DocMgr Administration** > **Alert Engine**. For more information configuring the alert e-mail server, see "Updating alert mail addresses" in *WebSphere Partner Gateway Partner Guide*.

## Viewing or editing alert details and contacts

### About this task

The internal partner can view all alerts, regardless of the Alert Owner (the creator of the alert).

1. Click **Account Admin** > **Alerts**. The system displays the Alert Search page.
2. Select the search criteria from the drop-down lists; enter the Alert Name. You can also click **Search** without selecting any search criteria (the system displays all alerts).
3. Click **Search**. The system displays the Alert Search Results page.
4. Click the View details icon to view an alert's details.
5. Click the Edit icon to edit alert details.
6. Edit information as required.
7. Click the **Notify** tab.
8. Select a partner (internal partner or hub administrator only). The internal partner can view all alerts regardless of the Alert Owner.
9. Edit contacts for this alert, if wanted.
10. Click **Save**.

## Searching for alerts

### About this task

1. Click **Account Admin** > **Alerts**. The system displays the Alert Search page.
2. Select the search criteria from the drop-down lists; enter the Alert Name. You can also click **Search** without selecting any search criteria (the system displays all alerts).

*Table 33. Alert search criteria for Partners*

| Value | Description |
|---|---|
| Alert Type | Volume, event, or all alert types. |
| Alert Name | Name of alert. |
| Alert Status | Alerts that are enabled, disabled, or all. |

*Table 33. Alert search criteria for Partners (continued)*

| Value | Description |
|---|---|
| Subscribed Contacts | Alert's assigned contacts. Selections are Has Subscribers, No Subscribers, or All. |
| Results Per Page | Controls how search results are displayed. |

*Table 34. Alert search criteria for internal partner and hub administrator*

| Value | Description |
|---|---|
| Alert Owner | Creator of the alert. |
| Alert Partner | Partner that the alert applies to. |
| Alert Type | Volume, event, or all alert types. |
| Alert Name | Name of alert. |
| Alert Status | Alerts that are enabled, disabled, or all. |
| Subscribed Contacts | Alert's assigned contacts. Selections are Has Subscribers, No Subscribers, or All. |
| Results Per Page | Controls how search results are displayed. |

3. Click **Search**. The system displays a list of alerts that meet your search criteria, if any.

## Disabling or enabling an alert

### Procedure

1. Click **Account Admin** > **Alerts**. The system displays the Alert Search page.
2. Select the search criteria from the drop-down lists; enter the Alert Name.
3. Click **Search**. The system displays a list of alerts that meet your search criteria, if any.
4. Locate the alert and click **Disabled** or **Enabled** under Status. Only the hub administrator and alert owner (creator of the alert) has permission to edit alert Status.

## Removing an alert

### Procedure

1. Click **Account Admin** > **Alerts**. The system displays the Alert Search page.
2. Select the search criteria from the drop-down lists; enter the **Alert Name**.
3. Click **Search**. The system displays a list of alerts that meet your search criteria, if any.
4. Locate the alert and click the Delete icon to delete. Only the hub administrator and alert owner (the creator of the alert) can remove an alert.

## Adding a new contact to an existing alert

### About this task

1. Click **Account Admin** > **Alerts**. The system displays the Alert Search page.
2. Enter the search criteria from the drop-down lists; enter the Alert Name.
3. Click **Search**. The system displays a list of alerts that meet your search criteria, if any.
4. Click the View details icon to view alert details.
5. Click the Edit icon to edit alert details.

6. Click the **Notify** tab.
7. Select a partner (internal partner and hub administrator only).
8. If the contact that you want to add is listed in the Contacts text box, select the contact and click **Subscribe**. Go to 13.

   If the contact that you want to add is not listed in the Contacts text box, click **Add New Entry to Contacts**. The system displays the Create New Contact pop-up window.

   Note that the Add New Entry to Contacts link is available only if the partner is hub operator.
9. Enter the contact's name, e-mail address, telephone and fax numbers.
10. Select the contact's Alert Status.
    - Select **Enabled** to begin sending e-mail messages to this contact when the system generates this alert.
    - Select **Disabled** if you do not want to send e-mail messages to this contact when the system generates this alert.
11. Select the contact's visibility.
    - Select **Local** to make the contact only visible to your organization.
    - Select **Global** to make the contact visible to the hub administrator and internal partner. Both of these parties can subscribe the contact to alerts.
12. Click **Save** to save the contact. Click **Save and Subscribe** to save the contact and add the contact to the list of contacts for this alert.
13. Click **Save.**

## Creating a volume-based alert

### About this task

1. Click **Account Admin** > **Alerts**. The system displays the Alert Search page.
2. Click **Create** in the upper right corner of the page. The system displays the Alerts Define tab.
3. Select **Volume Alert** for **Alert Type** (this is the default setting). The system displays the appropriate text boxes for a volume alert.
4. Enter an **Alert Name** for the alert
5. Enter **Custom business text**. When the alert event is generated, this message will be sent along with it.
6. Select an **Alert Owner** for the alert.
7. Select a **Partner** with rights to create a volume-based alert (internal partner and hub administrator only).
8. Select **Package**, **Protocol**, and **Document Type** from the drop-down lists. The selected Package, Protocol, and Document Type must match the Package, Protocol, and Document Type of the source external partner.
9. Select one of three volume options (Expected, Range, or Zero Volume), then proceed to 10 on page 281:
   - **Expected** - Select Expected if you want an alert generated when document type volume deviates from an exact quantity. Use the following steps to create an alert on expected document type volume:
     a. In the Volume text box, enter the number of document types you expect to receive within a time frame selected in 10 on page 281. Enter a positive number only; the alert will not function if you enter a negative number.

b. In the Percent Deviation text box, enter a number that defines the limit the document type volume can deviate from before the alert is activated. For example:
   - If Volume = 20 and Percent Deviation = 10, a document flow volume less than 18 or greater then 22 will trigger an alert.
   - If Volume = 20 and Percent Deviation = 0, any document flow volume other than 20 will trigger an alert.

- **Range**. Select Range to generate an alert if document flow volume falls outside a minimum-maximum range. Use the following steps to create an alert based on a range of values:
  a. In the Min text box, enter the minimum number of document flows you expect to receive within a time frame selected in 10. An alert is triggered only if document flow volume falls below this amount.
  b. In the Max text box, enter the maximum number of document flows you expect to receive within a time frame selected in 10.

  **Note:** Both Min and Max text boxes must be filled in when creating an alert based on volume range.
- **Zero Volume**. Select Zero Volume to trigger an alert if no document flows occur within a time frame selected in 10.

10. Select either Daily or Range for the time frame (Frequency) that the system will use to monitor document flow volume for alert generation.
    - **Daily**. Select Daily to monitor document flow volume on one or more actual days of the week or month. For example, select Daily if you are going to monitor document flow volume only on one or more specific days of the week (for example, Mondays, or Mondays and Thursdays), or month (for example, the 1st and the 15th).
    - **Range**. Select Range to monitor document flow volume between two days of the week or month. For example, select Range to monitor document flow volume on all days between Monday and Friday, or all days between the 5th and 20th of each month.
11. Select the **Starting** and **Ending time** (24-hour day) that the system will monitor document flow volume for the days selected in the next step. Note that when a Range frequency is selected, the document flow volume is monitored from the Starting time of the first day of the range through the Ending time on the last day of the range.
12. Select the appropriate days during the week or month that alert monitoring will occur. If you selected Daily as a frequency, select either the actual days of the week or days of the month for alert monitoring. If you selected Range as a frequency, select two days during the week, or two days during the month that alert monitoring will fall between.
13. Select the **Alert Status** of this alert as Enabled or Disabled.
14. Click **Save**.
15. Click the **Notify** tab.
16. Click the **Edit** icon.
17. Select a **partner** (internal partner and hub administrator only).
18. If the contact that you want to add is listed in the Contacts text box, select the contact and click **Subscribe**. Go to 23 on page 282.

    If the contact that you want to add is not listed in the Contacts text box, click **Add New Entry to Contacts**. The system displays the Create New Contact pop-up window.

Note that the Add New Entry to Contacts option is only presented to the Alert Owner to create contacts associated with the Alert Owner. This feature does not allow the Alert Owner to add contacts for Alert partners.

19. Enter the **contact's name**, **e-mail address**, **telephone** and **fax numbers**.
20. Select the contact's **Alert Status**.
    - Select **Enabled** to begin sending e-mail messages to this contact when the system generates this alert.
    - Select **Disabled** if you do not want to send e-mail messages to this contact when the system generates this alert.
21. Select the contact's visibility.
    - Select **Local** to make the contact only visible to your organization.
    - Select **Global** to make the contact visible to the hub administrator and internal partner. Both of these parties can subscribe the contact to alerts.
22. Click **Save** to save the contact; click **Save & Subscribe** to add the contact to the list of contacts for this alert.
23. Click **Save**.

**Note:** Changes made to volume-based alerts, after the original monitoring period, become effective on the next monitoring period day. For example, an alert monitors from 1-3 PM on Wednesdays and Thursdays. On Wednesday at 4 PM, the alert is changed to monitor from 5-7 PM. The alert will not monitor twice on Wednesday; the change will become effective on Thursday.

## Creating an event-based alert

### About this task

1. Click **Account Admin** > **Alerts**. The system displays the Alert Search page.
2. Click **Create** in the upper right corner of the page. The system displays the Alerts Define tab.
3. Select **Event Alert** for **Alert Type**. The system displays the appropriate text boxes for an event-based alert.
4. Enter an **Alert Name** for the alert.
5. Enter **Custom business text**. When the alert event is generated, this message will be sent along with it.
6. Select an **Alert Owner** for the alert.
7. Select a **Partner** that will trigger the alert (this option is only available to the internal partner and hub administrator). Select the Any Partner option to associate the alert with all the partners in the system. When you perform an alert search and select Any partner as the Alert Partner, the system displays all alerts that are not associated with a specific partner.
8. Select the **Event Type**: Debug, Information, Warning, Error, Critical, or All.
9. Select the **Event Name** that will activate the alert, for example, BCG240601 AS Retry Failure, or 108001 Not a Certificate. To create an alert that notifies you when a certificate is about to expire, select one of the following:
    - BCG108005 Certificate Expiration in 60 Days
    - BCG108006 Certificate Expiration in 30 Days
    - BCG108007 Certificate Expiration in 15 Days
    - BCG108008 Certificate Expiration in 7 Days
    - BCG108009 Certificate Expiration in 2 Days

**Note:** In order for an event to be listed here, it must be alertable. To make an Event alertable, see "Specifying alertable events" on page 289.

10. Select the status of this alert: Enabled or Disabled.
11. Click **Save**.
12. Click the **Notify** tab.
13. Select the **Notification mode**: Notify All Related Parties or Notify Subscribed Contacts Only. The subscribed contacts are notified by ***Notify Subscribed Contacts Only*** mode. While creating alerts, if the alert notification mode is selected as ***Notify All Related Parties***, then notification is sent to all the parties related to that event for which the alert is defined. The related parties for the event are the combined contacts of Source Participant, Target Participant and the Alert Owner.
14. Select a **Partner** (internal partner and hub administrator only).
15. From the contacts listed in the **Contacts** text box, select the contact you want notified, and click **Subscribe**.
16. Select the Mode of Delivery:
    - **Send alerts immediately**. When you select this option, the system sends alert notifications to the contact when the alert occurs. Use this option for critical alerts.
    - **Batch Alerts By**. When you select this option, you can specify when you want the contact to receive alert notifications. Use this option for non-critical alerts.

      The two options in this section, Count and Time, are not mutually exclusive.

      If you select the **Count** option, you must always select the Time option.

      – If the number of alerts (Count) is reached during the time limit that you have selected (Time), the system generates an alert notification.
      – If an alert occurs but the number of alerts (Count) is not reached during the time limit that you have selected (Time), the system will generate an alert notification at the end of the time limit.

      The **Time** option can be used without the Count option, but the Count option must always be associated with a time limit (Time).

      – **Count**. Must also use Time option when you select this option. Enter a number (n). This is the number of alerts that must occur during the selected time period (Time) before the system will send an alert notification to the alert's contact.

        Here's an example of how these two options work together:

        In our example, Batch Alerts By options are set to 10 for Count (10 alerts) and 2 for Time (2 hour period). The system retains all notifications for this alert until 10 occur in a two hour period or until the end of the time period is reached.

        When the alert count reaches 10 in a 2 hour period, the system sends all alert notifications for this alert to the contact.

        If an alert occurs but 10 alerts do not occur during the time limit (two hours), the system will send an alert notification to the alert's contact at the end of the time limit.
      – **Time**. Select number of hours (n). The system retains alert notification for n hours. Every n hours, the system sends all retained alert notifications to the contact.

For example, if you enter 2, the system retains all notifications for this alert that occur in each two hour interval. When the two hour interval expires, the system sends all alert notifications for this alert to the contact.

17. Click **Save**.

# Chapter 15. Initiating error flow

In WebSphere Partner Gateway, as an administrator, you can monitor failed events that occur while processing documents. A document can fail at the receiver or document manager. For a failed document, corresponding error or critical event is logged in the Event Engine. Alerts can be created to send e-mail notification to one or more subscribers.

Additionally, an administrator can actually initiate an error document flow for internal, external or both the partners. This error document will be initiated for a failed document based on the error or critical event. This error document flow can be either in WebSphere partner Gateway format or in Webservice format. You can configure the format in Error Flow configuration for an event.

## Error flow document configuration

### About this task

The Error Flow tab in the console allows the operator to set the Error Flow or Web Service invocation for certain error events:

1. Navigate to **Account Admin** > **Error Flow** tab. The error flow list has view and delete icons for each error flow.
2. Click **View** icon to launch the error flow configuration screen in read-only mode.
3. In the view configuration, click **Edit** icon to edit the error flow configuration.
4. In the edit mode, the following configuration values are available:
   - **Name** - error flow document configuration name.
   - **Sending Partner** - Click on partner search and select the partner name. The partner can be internal or external partner.
   - **Partner Type**- Select the partner type from the drop-down.
   - **Error Event** - this drop-down lists only events that are of *Error* or *Critical* type.
   - **Error Flow Type** - this can be *Error Flow Document* or *Invoke a Web Service*.
   - **Send To** - Select the recipients of the failed document. It can be *Sender* or *Receiver* or *Both*.
5. Click **Save**.
6. Click **Cancel** to cancel.
7. Enable B2B capabilities for the error flow configured.
8. If the Web service is invoked, then create the interaction and activate the Participant connection.

Error Flow document definitions for XML and WebService are uploaded by default in WebSphere Partner Gateway. You can enable these for partners and create the following connections:

- ErrorFlowDocument XML connection.
- ErrorFlowDocument over Webservices for document style.
- ErrorFlowDocument over Webservices for RPC style.

## Limitations and Restrictions

1. Error flow document over Web services have the following limitations:
   - Web services request must be one way request.
   - If the binding style is **document**, then the input parameter type is of element **ErrorFlowDocument** that is defined in BCGErrorFlowSchema.xsd.
   - If the binding style is **rpc**, then the input parameter type will be **String** and number of input parameters are one.
2. Error Flow Routing will not work in case of wrong Business IDs. If the error flow document is requested for a particular event and even if the Business document having improper IDs fails with the same configured event, then the error flow document routing will not work since the Business IDs specified are not valid.

# Chapter 16. Finishing the configuration

This chapter describes additional tasks you can perform to configure the hub. It includes the following topics:



**Note:** you should always use the same browser instance with which you logged into the Community Console to make configuration changes to WebSphere Partner Gateway. Using more than one browser instance at the same time can result in nullifying your configuration changes.

## Large file support for AS documents

Large file support with an order of size in GBs has been extended for AS2, and AS3. The maximum file size processed using byte arrays is configurable. When the amount of memory allocated is more than the available heap size, OutOfMemoryError occurs. If the size of data is less than the available memory, OutOfMemoryError may still occur if the memory allocated increases available memory. At runtime it is determined whether the file size configured can be supported based on available heap memory. You can specify the maximum file size that can be used with byte arrays using the property **bcg.maximumFileSizeForByteArrays**. The value of the property bcg.maximumFileSizeForByteArrays is in MBs. If the file size is more than the value of this property, it is processed using streams. If the file size is less than the value of this property, and if sufficient memory is not available, an error event BCG210050 is generated.

When you log in as a hub operator, navigate to **System Administration** tab > **Common Properties** tab. Overwrite the default value of **bcg.maximumFileSizeForByteArrays** property to specify the maximum file size to be used with byte arrays. Increase the value of this property for better performance.

## Enabling the use of APIs

### About this task

WebSphere Partner Gateway supplies a set of APIs that can be used to access certain functions typically performed on the Community Console. These APIs are described in the *WebSphere Partner Gateway Programmer Guide*.

Use this procedure to enable the use of the XML-based APIs so that partners can make API calls to the WebSphere Partner Gateway server.

### Procedure

1. From the main menu, click **System Administration > Feature Administration > Administration API**.
2. Click the **Edit** icon next to **Enable the XML-Based API**.
3. Select the check box to enable the use of the XML-based API.
4. Click **Save**.

### Results

**Note:** The XML-based administrative API is deprecated.

The migration utility can also be used instead of the administrative API to perform the create and update tasks. The migration import file has new or updated information.

The import file is described by the XML schema that is provided with the migration utility. You can use a development tool such as Rational Application Developer to produce an import XML file that conforms to the schema. By importing this file with the migration utility, you can load new partner definitions including contacts and business IDs for the partners. You can also update existing partner definitions by importing them with the migration utility. The administrative API also allows you to list some of the configuration artifacts in a system. A full export of the system using the migration utility provides listings of partner capabilities, partner connections, and receivers (targets) in the exported xml file.

The **bcgmigrate.bat/bcgmigrate.sh** batch file is used to initiate the migration process. While running the bcgmigrate command, ensure that you have **Execute** file permission for (bcgmigrate.bat/bcgmigrate.sh). This is more applicable for UNIX platform.

## Specifying the queues used for events

### About this task

You can configure the hub to deliver events to an external queue that is configured using JMS configuration.

The default JMS configuration is established when you install the hub. You can see some of these values on the Event Publishing Properties page.

To point to a different JMS configuration, provide appropriate configuration values for either publishing the events to WebSphere Partner Gateway / WAS internal messaging queues or other messaging servers. Also, change the queue name to match the name of the queue where the events are published.

To indicate where events should be delivered:

1. From the main menu, click **System Administration > DocMgr Administration > Event Engine > External Events**.
2. Click the **Edit** icon next to **Enable Event Delivery**.
3. Select the **Enable Event Delivery** check box to activate event publishing.
4. If the default values are correct for your installation, leave them as is. The default values support event delivery to the queue named DeliveryQ provided by the JMS Server that you configured at installation.

If you want to change where events are delivered, update the fields, using the following information as reference:

- Enter values for **User ID** and **Password**, if a user ID and password are required to access the queue
- For **JMS Queue Factory Name**, enter the name of the JMS Queue Connection Factory from the JMS .bindings file that you are using.

  **Note:** On some Windows versions (prior to XP), you might need to change the default value of the **JMS Queue Factory Name** field if you want to use the default Event Delivery feature. You will need to change the value for **JMS Queue Factory Name** from: `WBIC/QCF` to `WBIC\\QCF`.
- For **JMS Message Type**, enter the type of message that will be delivered. The choices are byte or text. As the Receiver component decides the JMS message type mapping, the value of JMS Message type is optional.
- For **JMS Queue Name**, enter the name of the JMS queue to which the events will be published. This queue must already be defined in the JMS .bindings file that you are using in WebSphere MQ.

  **Note:** On some Windows versions (prior to XP), you might need to change the default value of the **JMS Queue Name** field if you want to use the default Event Delivery feature. You will need to change the value for **JMS Queue Name** from `WBIC/DeliveryQ` to `WBIC\\DeliveryQ. WBIC/QCF`.
- For **JNDI Factory Name**, enter the name used to access the .bindings file. The default value provides access to the default binding in the file system.
- For **Provider URL Packages**, enter a URL that provides access to the JMS bindings file. This URL must be consistent with the JNDI Factory Name. This field is optional and, when not filled in, it uses the default file system location for JMS bindings.
- For **Message Char Set**, enter the character set to be used when creating the byte message on the JMS queue. The default value is UTF-8. This field is relevant only for byte messages.
- For **JMS Provider URL**, enter the URL of the JMS provider. This field is optional and when not filled in, it uses the default JMS provider that was identified at installation.

5. Click **Save**.

# Specifying alertable events

## About this task

When an event occurs within WebSphere Partner Gateway, an event code is generated. Using the Event Codes page, you can set the alertable status of the event code. When an event is set as alertable, the event appears in the Event Name list of the Alert page. You can then set an alert for the event.

To indicate which events should be alertable:

## Procedure

1. Click **Hub Admin > Hub Configuration > Event Codes**. The Event Codes page is displayed.
2. For each event you want made alertable:
   a. Click the **View details** icon next to the event code. The Event Code Details page is displayed.

b. Select **Alertable**.
c. Click **Save**.

## Updating a user-defined transport

As described in Chapter 7, "Defining receivers," on page 55 and Chapter 11, "Creating destinations," on page 209, you can upload an XML file that describes a user-defined transport. You use **Manage Transport Types** to upload the file. After you upload the XML file, the transport becomes available for use when defining a receiver or destination.

The XML file that describes the user-defined transport includes the attributes for the transport. These attributes are displayed (in the section **Custom Transport Attributes**) on the receiver or destination page when you specify a user-defined transport. For example, a user-defined transport for a destination might include the attribute DestinationRetryCount.

The person who wrote the XML file describing the transport can update the attributes (by adding, deleting, or modifying the attributes). If the XML file is modified, you again use **Manage Transport Types** to upload the file. Any changes to the attributes are reflected in the destination or receiver page.

## Samples

WebSphere Partner Gateway consists is packaged with few samples, which provides custom functionality and illustrations. These packages are found in the directory where WebSphere Partner Gateway install is extracted, under the folders **DevelopmentKits** and **Integration**.

The DevelopmentKits folder contains the following samples:

- Administrative API: as Administrative APIs are deprecated, the Partner Migration utility is used to create and update tasks.
- Migration: contains samples for Export and Import configuration.
  - Export configuration: illustrates the procedure to export the WebSphere Partner Gateway configurations using a java component from the command line script file.
  - Import configuration: illustrates the procedure to import the WebSphere Partner Gateway configurations using a java component from the command line script file.
- UserExits: consists of samples for writing custom user exit code for translation and validation.
  - The *EDITransTypeBusinessProcess* sample provides custom functionality for EDI Documents that are passing through the system. This sample user exit is designed to parse the EDI Transaction Type from an EDI X12 document. By modifying the parsing criteria, other values can be extracted.
  - The *custom translation user exit* sample provides translation functionality for an inbound XML document.
  - The *custom validation user exit* sample provides validation functionality for an inbound XML document.
- Sample Scenarios: consists of samples that provides the guidelines for setting up a WebSphere Partner Gateway System for the protocols mentioned below, with No packaging as well AS packaging. For each protocol, the configuration import file is also provided.

- Custom XML
- EDI-X12
- Binary documents

The Integration folder contains the following integration samples:

- WebSphere Transformation Extender integration : sample to demonstrate integration with WebSphere Transformation Extender for transforming an XML document to flat file.
- WebSphere Business Integration Message Broker sample : sample to demonstrate how WebSphere Partner Gateway communicates with WebSphere Business Integration Message Broker.
- WebSphere Process Server integration: sample to demonstrate how WebSphere Partner Gateway integrates with WebSphere Process Server over JMS.
- WebSphere Interchange Server integration: sample to demonstrate how WebSphere Partner Gateway integrates with Interchange Server using HTTP and JMS.

# Chapter 17. CPP/CPA Editor

CPP/CPA editor is a eclipse plugin that assists in creating CPP/CPA document from template and allows the user to edit using table format. Also, it handles data and schema validation.

**Prerequisites:**

- WID/RAD versions 6.1 and above is required
- Place the downloaded CPP/A editor plug-in in the plug-in folder of IDE

A Collaboration-Protocol Agreement (CPA) document can also be created from two Collaboration-Protocol Profile (CPP) documents. CPP defines the capabilities of a party engaged in an electronic Business with other parties. CPA describes the Message-exchange agreement between two Parties. To create a CPP, enter the values for individual XML elements (individual XML elements are composed of various attributes) through the User Interface of the editor. Once the CPA document is created using the editor and its status is "AGREED", it can be imported in WebSphere Partner Gateway. The imported files automatically create the following:

- Partners
- B2B Gateways
- Interactions and connections

Apart from that, it automatically defines the document definitions and enables the required B2B capabilities.

You can do the following using the user interface of the CPP/CPA editor:

- "Creating CPP document"
- "Creating CPA document" on page 294
- "Editing values in the editor" on page 294

To make CPP/CPA editor as the default editor, do the following:

1. In the eclispe plugin environment, click **Window** menu and select **Preferences**
2. In preferences window, click **General > Editor > File Association**.
3. Select "*.xml" in **File types** list and "CPPEditor Multi – page Editor" in **Associated editors** list.
4. Click **Default**.

## Creating CPP document

To create a CPP document, do the following:

1. In the IDE, select **File** >**New**.
2. In the **New** window, select **CPAEditor** > **Collaboration Protocol Profile file**
3. Click **Next** and enter the CPP/CPA holder values.
4. Click **Finish**. The new file gets created under the specified container.
5. If you have configured CPAEditor as your default, then modify the values in the template. Otherwise, the file will open in XML editor. To open the file in CPAEditor, right-click and select **Open with** > **CPAEditor Multi-Page Editor**.

6. Enter the values for attributes of all elements. For certain attributes, you can select the appropriate value from the different options.
7. Click **Save**. A message is displayed confirming the successful creation of a CPP document.

## Creating CPA document

You need to select one of the following two options:

- Case 1: Creating a CPA using a template, allows entering of values for individual XML elements (individual XML elements are composed of various attributes) through the User Interface of the editor.
- Case 2: Creating a CPA from two CPPs

To create a CPA using template, do the following:

1. In the IDE, select **File** >**New**.
2. In the **New** window, select **CPAEditor** > **Collaboration Protocol Agreement file**
3. Click **Next** and enter the CPP/CPA holder values.
4. Click **Finish**. The new file gets created under the specified container.
5. If you have configured CPAEditor as your default, then modify the values in the template. Otherwise, the file will open in XML editor. To open the file in CPAEditor, right-click and select **Open with** > **CPPEditor Multi-Page Editor**.
6. Enter the values for attributes of all elements. For certain attributes, you can select the appropriate value from the different options.
7. Click **Save**. A message is displayed confirming the successful creation of a CPA document.

To create a CPA from two CPPs, do the following:

1. In the IDE, click **File** > **New** > **Other**.
2. In the **New** window, select **CPAEditor** > **Merge Collaboration Protocol Profiles**.
3. Click **Next**
4. Enter the CPP/CPA holder values and path and names of the CPP files that you want to merge.
5. Click **Finish**. The merged file gets created in the specified container.
6. If you have configured CPAEditor as your default, then modify the values in the template. Otherwise, the file will open in XML editor. To open the file in CPAEditor, right-click and select **Open with** > **CPPEditor Multi-Page Editor**.

## Editing values in the editor

To edit the values in the editor table, place the cursor on the cell and edit the values. Every PartyInfo element has a unique partyName associated with it. The various sub-elements that occur under PartyInfo are PartyId, PartyRef, Collaboration Role, Certificate, SecurityDetails, DeliveryChannel, Transport, DocExchange, and OverrideMshActionBinding. These values are available in different tabs in the CPP/CPA editor. PartyName serves as a unique identifier to associate the sub elements of the PartyInfo with the corresponding PartyInfo element.

For example, Certificate element that is the sub-element of PartyInfo element can occur one or more times. The PartyInfo element can itself occur multiple times in a CPP.

# Chapter 18. Web Mail Box

The new features provided in Web Mail Box release is an addition to the existing support of WebSphere Partner Gateway. This allows partners, customers, and vendors to interact with the hub using only the supported browsers, that is, web based support for B2B interaction. The web version of the WebSphere Partner Gateway Console is opened in a browser and no external infrastructure is required, such as FTP Server, email facility, and so on. The following additional tasks can be performed in this version of WebSphere Partner Gateway:

- Upload documents to transact
- Monitor the status of business documents
- Download the received business document

This feature is primarily for external partners, who do not have the infrastructure to participate in the transactions. This chapter provides the necessary prerequisite steps, which are required to work with Web Mail Box feature.

**Note:** This release supports documents only in "None" package.

## Prerequisites

For an external partner to make use of the Web Mail Box features, the Hub administrator has to provide the following permissions:

- "Enabling Web Mail Box at Hub level"
- "Enabling Web Mail Box at partner level"
- "Enabling WebBoxReceiver" on page 298

### Enabling Web Mail Box at Hub level

#### About this task

To enable permissions for Inbox and Outbox:

1. Navigate to **Hub admin** > **Console config** > **Permissions** page.
2. In the **Permission** list page, enable Inbox and Outbox.

**Note:** This is a one time activity for Hub admin.

### Enabling Web Mail Box at partner level

#### About this task

The steps to be performed for a new external partner:

1. Login to the console as a hub administrator.

   **Note:** When you create a new external partner, a default Webuser group is created automatically. Likewise, when this patch is installed, default Webuser group is created for existing partners.
2. In the **Groups** page, click view permissions icon for the newly created group.
3. Select **Read / Write** for Inbox and Outbox.
4. Create a new User.
5. In the **Memberships** page, assign the user to the group.

**Note:** This is a one time activity for Hub admin.

## Enabling WebBoxReceiver

### About this task

After the installation of the web mailbox feature, the Hub administrator has to enable the receiver prior to sending documents to internal partner. The default state of WebBoxReceiver is disabled.

**Note:** WebBoxReceiver is system created and cannot be deleted.
The steps to enable WebBoxReceiver are as follows:

1. Navigate to **Hub admin** > **Receivers**.
2. Enable WebBoxReceiver.

**Note:** To modify the poll interval of WebBoxReceiver, edit its poll interval attribute accordingly.

---

# Web Mail Box limitations

The following are the limitations of Web Mail Box:

- Can send a Maximum of 10 MB to internal partners.

  **Note:** It can be more or less based on the network, browser, and memory.
- Cannot delete Web box receiver.
- Cannot send EDI/XML documents in Binary format.

# Chapter 19. Basic examples

This appendix provides examples of configuring the hub. It includes the following topics:

- " Basic Configuration – Exchanging passthrough EDI documents"
- " Basic configuration - Setting up security for inbound and outbound documents" on page 305
- "Extending the basic configuration" on page 310

A separate appendix is provided for examples of exchanging EDI interchanges that include de-enveloping, transformation, enveloping, and functional acknowledgment transmission. See Chapter 20, "EDI examples," on page 317.

These examples are intended to provide you with a quick overview of the steps required to configure a system. If you are using these examples to set up your system, modify the specific information (for example, names and business IDs) to suit the needs of your business.

## Basic Configuration – Exchanging passthrough EDI documents

In this example, the hub configuration is quite simple—two receivers are defined (one for documents coming into the hub from a partner and one for documents coming into the hub from the internal partner back-end system). The exchanges that are set up in this example use the document definitions provided by WebSphere Partner Gateway; therefore, you only have to create interactions based on those flows. No custom XML is used in this example.

This example shows an exchange between a back-end-application of the internal partner and an external partner (Partner Two).

### Configuring the hub

The first step in setting up the hub is creating the two receivers.

- An HTTP Receiver (called "HttpReceiver") to receive documents over HTTP (from Partner Two) that are to be sent to the back-end system of the internal partner.
- A File Directory Receiver (called "FileSystemReceiver") to retrieve documents from the file system (from the internal partner's back-end system) that are to be sent to Partner Two) .

#### Defining the receivers

**About this task**

To create a receiver for the receipt of documents over HTTP:

1. Click **Hub Admin > Hub Configuration > Receivers**.
2. Click **Create Receiver**.
3. For Receiver Name, type: **HttpReceiver**.
4. From the Transport list, select **HTTP/S**.
5. For the Operation mode, use the default of **Production**.
6. For the URI, type: **/bcgreceiver/submit**

7. Click **Save**.

Next, you create a receiver to poll a directory on the file system. Creating the receiver automatically creates a new directory on the file system.

To create the file-system receiver:

1. Click **Hub Admin > Hub Configuration > Receivers**.
2. Click **Create Receiver**.
3. For Receiver Name, type: **FileSystemReceiver**.
4. From the Transport list, select **File Directory**.
5. For Default Operation mode, use the default of **Production**.
6. For the Document Root Path, type: **\temp\FileSystemReceiver**

   **Note:** This will create a FileSystemReceiver directory within the temp directory. Be sure a temp directory exists on the file system.
7. Click **Save**.

## Defining document types and interactions

### About this task

In this example, you are setting up the exchange of documents that conform to the EDI-X12 standard. In this example, the documents are simply being passed through the hub. The EDI interchange is not being de-enveloped and no transformation occurs. See Chapter 23, "Attributes," on page 409 for examples of de-enveloping an interchange, transforming the transactions, and sending acknowledgments.

In this section, the following exchanges are described:

- Sending an EDI-X12 document, with no packaging, from the internal partner to Partner Two
- Sending an EDI-X12 document, packaged in AS2, from Partner Two to the internal partner

Because of the packaging and protocols involved, there is no need to create a new document definition. The packages, protocols, and document types are ones that are predefined in the system.

However, you do need to define interactions based on these predefined document types.

Create the first interaction, in which the source is an ISA-formatted document that conforms to the EDI-X12 standard and contains no packaging and the target is an ISA-formatted document that conforms to the EDI-X12 standard with AS packaging.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. From the **Source** column, expand:
   a. **Package: None**
   b. **Protocol: EDI-X12**
4. Click **Document Type: ISA**
5. From the **Target** column, expand:

a. **Package: AS**
b. **Protocol: EDI-X12**

6. Click **Document Type: ISA**
7. From the **Action** list, select **Pass Through**.
8. Click **Save**.

Create a second interaction, in which the source format is an ISA-formatted document that conforms to the EDI-X12 standard with AS packaging, and the target format is an ISA-formatted document that conforms to the EDI-X12 standard and contains no packaging:

1. Click **Create Interaction link**.
2. From the **Source** column, expand:
   a. **Package: AS**
   b. **Protocol: EDI-X12**
3. Click **Document Type: ISA**
4. From the **Target** column, expand:
   a. **Package: None**
   b. **Protocol: EDI-X12**
5. Click **Document Type: ISA**
6. From the **Action** list, select **Pass Through**.
7. Click **Save**.

## Creating partners and partner connections

In this example, one external partner is created, in addition to the internal partner. The destinations for the partners include standard transports, and no configuration points are defined for the destinations.

### Creating the partners

Create two new partners. To define the internal partner:

1. Click **Account Admin** from the main menu. The Partner Search page is the default view.
2. Click **Create**.
3. For **Company Login Name**, type: **CommMan**.
4. For **Partner Display Name**, type: **Comm Man**.
5. For **Partner Type**, select **Internal Partner**.
6. Click **New** under **Business ID**.
7. Leave **Type** as **DUNS** and enter an Identifier value of **123456789**.

   **Note:** Here and throughout this book, all DUNS numbers are meant to be examples only.
8. Click **New** under **Business ID**.
9. Select **Freeform** and enter an Identifier value of **12-3456789**
10. Click **Save**.

To define Partner Two:

1. Click **Account Admin > Profiles > Partner**.
2. Click **Create**.

3. For **Company Login Name**, type: **partnerTwo**
4. For **Partner Display Name**, type: **Partner Two**
5. For **Partner Type**, select **External Partner.**
6. Click **New** under **Business ID**.
7. Leave **Type** as **DUNS** and enter **987654321** as the Identifier.
8. Click **New** under **Business ID**.
9. Select **Freeform** and enter an Identifier value of **98-7654321**
10. Click **Save**.

You have now defined both the internal partner and Partner Two to the hub.

The next steps are to configure destinations for both the internal partner and Partner Two.

## Creating the destinations

### About this task

Before creating a file-directory destination for the internal partner, you must create the directory structure used by this destination. Create a new FileSystemDestination directory on the root drive. This directory will be used by the internal partner to store files received from external partners.

In the case of the internal partner, the destination represents the entrance point into the back-end system.

To create a destination for the internal partner:

1. Click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Select **Internal Partner** by clicking the **View details** icon.
4. Click **Destinations** from the horizontal navigation bar.
5. Click **Create**.
6. For **Destination Name**, type: **FileSystemDestination**.
7. For **Transport**, select **File Directory**.
8. For **Address**, type: **file://C:\FileSystemDestination**.
9. Click **Save**.

Next, set this newly created destination as the default destination for the internal partner.

1. Click **List** to view all destinations configured for the internal partner.
2. Click **View Default Destinations**.
3. From the **Production** list, select **FileSystemDestination**.
4. Click **Save**.

Create a destination for Partner Two

1. Click **Account Admin > Profiles > Partner**.
2. Click **Search**, and then select **Partner Two** by clicking the **View details** icon.
3. Click **Destinations** from the horizontal navigation bar.
4. Click **Create**.
5. For **Destination Name**, type: **HttpDestination**.
6. For **Transport**, select **HTTP/1.1**.

7. For **Address**, type: **http://<***IP_address***>:80/input/AS2**, where <*IP_address*> represents Partner Two's computer.
8. For **User Name**, type: **Comm Man**.
9. For **Password**, type: **commMan**.
10. Click **Save**.

Note that this example assumes that Partner Two requires a user name and password for any partner logging in to its system.

Again, you need to define a default destination for this partner.

1. Click **List** followed by **View Default Destinations**.
2. From the **Production** list, select **HttpDestination**.
3. Click **Save**.

## Setting up B2B Capabilities

### About this task

Next, define the B2B Capabilities for the internal partner.

1. From the main menu, click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Select **Internal Partner** by clicking the **View details** icon.
4. Click **B2B Capabilities** from the horizontal navigation bar.
5. Set the Source and Target for Package: None, Protocol: EDI-X12, and Document Type: ISA by performing the following steps:
   a. Click the **Role is not active** icon under **Set Source** for **Package: None**
   b. Click the **Role is not active** icon under **Set Target** for **Package: None**
   c. Click the **Expand** icon next to **Package: None**.
   d. Click the **Role is not active** icon for **Protocol: EDI-X12 (ALL)** for both source and target.
   e. Click the **Expand** icon next to **Protocol: EDI-X12 (ALL)**.
   f. Click the **Role is not active** icon for **Document Type: ISA** for both source and target.

Then, set the B2B Capabilities for Partner Two.

### Procedure

1. From the main menu, click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Select Partner Two by clicking the **View details** icon.
4. Click **B2B Capabilities** from the horizontal navigation bar.
5. Select Set Source and Set Target for Package: AS, Protocol: EDI-X12, and Document Type: ISA by performing the following steps:
   a. Click the **Role is not active** icon under **Set Source** for **Package: AS**
   b. Click the **Role is not active** icon under **Set Target** for **Package: AS**
   c. Click the **Expand** icon next to **Package: AS**.
   d. Click the **Role is not active** icon for **Protocol: EDI-X12 (ALL)** for both source and target.
   e. Click the **Expand** icon next to **Protocol: EDI-X12 (ALL)**.

f. Click the **Role is not active** icon for **Document Type: ISA** for both source and target.

## Defining partner connections

### About this task

Define the partner connection for EDI documents with no packaging that come from the internal partner to be delivered to Partner Two.

1. Click **Account Admin > Connections**.
2. From the **Source** list, select **Internal Partner**.
3. From the **Target** list, select **Partner Two**.
4. Click **Search**.
5. Click **Activate** for the connection with the following detail:
   a. **Source**
      1) Package: **None (N/A)**
      2) Protocol: **EDI-X12 (ALL)**
      3) Document Type: **ISA(ALL)**
   b. **Target**
      1) Package: **AS (N/A)**
      2) Protocol: **EDI-X12 (ALL)**
      3) Document Type: **ISA (ALL)**

Next, define the connection for EDI documents wrapped in AS2 packaging that come from Partner Two to be delivered to the internal partner with no packaging. This is very similar to the connection you defined in the previous section, except that you will also configure AS2 attributes.

1. Click **Account Admin > Connections**.
2. From the **Source** list, select **Partner Two**
3. From the **Target** list, select **Internal Partner**.
4. Click **Search**.
5. Click **Activate** for the connection with the following detail:
   a. **Source**
      1) Package: **AS (N/A)**
      2) Protocol: **EDI-X12 (ALL)**
      3) Document Type: **ISA (ALL)**
   b. **Target**
      1) Package: **None (N/A)**
      2) Protocol: **EDI-X12 (ALL)**
      3) Document Type: **ISA (ALL)**

Next, select Attributes next to the **Package: AS (N/A)** box for Partner Two.

1. Edit the Package: AS (N\A) attributes by scrolling down the page and clicking the **Expand** icon next to **Package: AS (N/A)**.
2. Enter an AS MDN E-Mail Address (AS1) value. This can be any valid e-mail address.
3. Enter an AS MDN HTTP URL (AS2) value. This should be entered as follows: **http://<***IP_address***>:57080/bcgreceiver/submit,** where <*IP_Address*> represents the hub.
4. Click **Save**.

# Basic configuration - Setting up security for inbound and outbound documents

In this section, you will see how to add the following types of security to the basic configuration:

- Secure Socket Layers (SSL) Server Authentication
- Encryption
- Digital Signatures

## Setting up SSL authentication for incoming documents

### About this task

In this section, you use iKeyman to set up server authentication so that Partner Two can send AS2 documents over HTTPS.

To set up server authentication, perform the following steps:

1. Initiate the iKeyman application, by opening the ikeyman.bat file from the /<*ProductDir*>/was/bin directory.
2. Open the Receiver's default key store, bcgSecurity.jks. From the menu bar, select **Key Database File Open**. On a default installation, bcgSecurity.jks resides in the directory: <*ProductDir*>/common/security/keystore
3. When prompted, enter the default password for bcgSecurity.jks. This password is WebAS.
4. If this is the first time you have opened bcgSecurity.jks, delete the “Dummy” certificate.

The next step is to create a new self-signed certificate. Creating a self-signed personal certificate creates a private key and public key within the server key store file.

To create a new self-signed certificate:

1. Click **New Self Signed**.
2. Give the certificate a key label that is used to uniquely identify the certificate within the key store. Use the label **selfSignedCert**.
3. Enter the server’s Common Name. This is the primary, universal identity for the certificate. It should uniquely identify the principal that it represents.
4. Enter the name of your organization.
5. Accept all other defaults, and click **OK**.

Assume that Partner Two wants to send an EDI message over AS2 using secure HTTP. Partner Two will need to refer to the public certificate (which was created as part of the creation of the self-signed certificate) in order to do so.

To enable Partner Two to use the public certificate, export the public certificate from the server key store file as follows:

1. Select the newly created self-signed certificate from the IBM Key Management utility.
2. Click **Extract Certificate**.
3. Change the Data type to **Binary DER data**.
4. Provide the file name **commManPublic** and click **OK**.

Finally you use iKeyman to export the self-signed certificate and private key pair in the form of a PKCS12 file. This PCKS12 file will be used for encryption, which is described in a later section.

To export the self-signed certificate and private key pair:

1. Click **Export/Import**.
2. Change the Key file type to **PKCS12**.
3. Provide the File Name **commManPrivate** and click **OK**.
4. Enter a password to protect the target PKCS12 file. Confirm the password, and click **OK**.

**Note:** Stop and restart the Receiver for these changes to take effect.

The password entered will be used later when you import this private certificate into the hub.

Partner Two must also perform some configuration steps, including importing the certificate and changing the address to which it sends AS2 documents. For example, Partner Two would have to change the address to:

`https://<IP_address>:57443/bcgreceiver/submit`

where *<IP_address>* refers to the hub.

Now, the self-signed certificate that was placed in the Receiver's default key store is presented to Partner Two whenever Partner Two sends a document over secure HTTP.

To set up the reverse situation, Partner Two must provide the hub with an SSL key in the form of a .der file (in this case, partnerTwoSSL.der). If necessary, Partner Two must also change the configuration to permit the receipt of documents over the HTTPS transport.

Load Partner Two's file, partnerTwoSSL.der, into the Hub Operator's profile as a root certificate. A root certificate is a certificate issued from a Certifying Authority (CA) used when establishing a certificate chain. In this example, PartnerTwo generated the certificate, which is loaded as a root certificate to allow the hub to recognize and trust the sender.

Load partnerTwoSSL.der into the hub:

1. From the main menu, click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Select **Hub Operator** by selecting the **View details** icon.
4. Click **Certificates** and then **Load Certificate**.
5. Set the **Certificate Type** as **Root and Intermediate Certificate**.
6. Change the Description to **Partner Two SSL Certificate**.
7. Set the **Status** as **Enabled**.
8. Click **Browse** and navigate to the directory in which you have saved partnerTwoSSL.der.
9. Select the certificate and click **Open**.
10. Click **Upload** and then click **Save**.

Change Partner Two's destination to use secure HTTP.

1. Click **Account Admin > Profiles > Partner** from the horizontal navigation bar.
2. Click **Search** and select Partner Two by clicking the **View details** icon.
3. Click **Destinations** from the horizontal navigation bar. Next select HttpDestination by clicking the **View details** icon.
4. Edit it by clicking the **Edit** icon.
5. Change the transport value to **HTTPS/1.1**
6. Change the value of the address to read as follows: **https://<***IP_address***>:443/input/AS2**, where <*IP_address*> refers to Partner Two's machine.
7. All other values can remain unchanged. Click **Save**.

## Setting up encryption

### About this task

This section provides the steps for setting up encryption.

Partner Two must perform any necessary configuration steps (for example, importing the public certificate and the self-signed certificate) and set up encryption on documents sent to the hub.

WebSphere Partner Gateway will use its private key when decrypting documents. To allow the hub to do so, you first load the private key extracted from the self-signed certificate into the Community Console. Perform this task logged in to the Community Console as Hub Operator and install the certificate in your own profile.

To load the PKCS12 file:

1. Click **Account Admin > Profiles > Partner** from the horizontal navigation bar.
2. Click **Search**.
3. Select **Hub Operator** by clicking the **View details** icon.
4. Click **Certificates** and then click **Load PKCS12**.
5. Select the check box to the left of **Encryption**.
6. Change the Description to **CommManPrivate**.
7. Select **Enabled**.
8. Click **Browse** and navigate to the directory in which the PKCS12 file, commManPrivate.p12, is stored.
9. Select the file and click **Open**.
10. Enter the password provided for the PKCS12 file.
11. Leave the Operation mode as **Production**.
12. Click **Upload**, and then click **Save**.

This completes the configuration required to allow a partner to send encrypted transactions over secure HTTP to the hub.

In the following section, the previous procedure is reversed—the hub sends an encrypted EDI transaction over secure HTTP.

Partner Two must generate a document decryption key pair (in this example, partnerTwoDecrypt.der) and should make the public certificate available to the hub.

As mentioned earlier, the public key will be used by the hub when encrypting transactions to be sent to the partner. In order to do so, you load the public certificate into the hub.

### Procedure

1. From the main menu, click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Select Partner Two by clicking the **View details** icon.
4. Click **Certificates** from the horizontal navigation bar.
5. Click **Load Certificate**.
6. Select the check box next to **Encryption**.
7. Change the Description to read **Partner Two Decrypt**.
8. Set the status to **Enabled**.
9. Click **Browse**.
10. Navigate to the directory in which the decryption certificate, partnerTwoDecrypt.der, is stored.
11. Select the certificate and click **Open**.
12. Leave the Operation mode as **Production**
13. Click **Upload** and then click **Save**.

### Results

The final step in configuring the hub to send encrypted messages over secure HTTP using AS2 is to modify the partner connection that exists between the internal partner and Partner Two.

To modify the partner connection from the Community Console:

1. Click **Account Admin > Connections** from the horizontal navigation bar.
2. From the **Source** list, select **Comm Man**.
3. From the **Target** list, select **Partner Two**.
4. Click **Search**.
5. Click the **Attributes** button for the Target.
6. From the Connection Summary, note that the **AS Encrypted** attribute has a current value of **No**. Edit this value by clicking the **Expand** icon next to **Package: AS (N/A)**.

   **Note:** You will need to scroll down the page for this option to appear.
7. From the list, update the **AS Encrypted** attribute to **Yes** and click **Save**.

## Setting up document signing

### About this task

When digitally signing a transaction or message, WebSphere Partner Gateway uses your private key to create the signature and sign. Your partner receiving that message uses your public key to validate the signature. WebSphere Partner Gateway uses digital signatures to this effect.

This section provides the steps required to configure both the hub and a partner for use with digital signatures.

Partner Two must perform any necessary configuration steps (for example, creating a self-signed document named, in this example, partnerTwoSigning.der) and configuring the signing of documents. Partner Two must make partnerTwoSigning.der available to the hub.

To load the digital certificate into the hub:

1. Click **Account Admin > Profiles > Partner** from the horizontal navigation bar.
2. Click **Search**.
3. Select Partner Two by clicking the **View details** icon.
4. Choose **Certificates** from the horizontal navigation bar.
5. Click **Load Certificate**.
6. Select the check box next to **Digital Signature**.
7. Change the Description to **CommMan Signing**.
8. Set the **Status** to **Enabled**.
9. Click **Browse**.
10. Navigate to the directory in which the digital certificate, partnerTwoSigning.der, is saved, select the certificate, and click **Open**.
11. Click **Upload** followed by **Save**.

This completes the initial configuration for digital signatures.

The partner uses the public certificate to authenticate signed transactions sent the hub.

The hub will use the private key to digitally sign outbound transactions sent to the partner. You first enable the private key for digital signature.

To enable the private key for digital signature:

1. Click **Account Admin > Profiles > Certificates** from the horizontal navigation bar.
2. Click the **View details** icon next to **Hub Operator**.
3. Click the **View details** icon next to **CommManPrivate**.

   **Note:** This was the private certificate loaded into the hub earlier.
4. Click the **Edit** icon.
5. Select the check box next to **Digital Signature**.

   **Note:** If there were more than one digital signature certificate, you would select which one was primary and which one was secondary by selecting **Primary** or **Secondary** from the **Certificate Usage** list.
6. Click **Save**.

Next you alter the attributes of the existing partner connection between the internal partner and Partner Two to accommodate signed AS2.

To alter the attributes of the partner connection:

1. Click **Account Admin > Connections** from the horizontal navigation bar.
2. Select **Internal Partner** from the **Source** list.
3. Select **Partner Two** from the **Target** list.
4. Click **Search**.

5. Click the **Attributes** button for Partner Two.
6. Edit the **AS Signed** attribute by clicking the **Expand** icon next to **Package: AS (N/A)**.
7. Select **Yes** from the **AS Signed** list.
8. Click **Save**.

This completes the configuration required to send a signed AS2 transaction from WebSphere Partner Gateway to the partner.

# Extending the basic configuration

This section shows you how to modify the basic configuration described in this appendix. Using the same partners and setup described earlier (an internal partner, using a DUNS ID of 123456789 and a file-directory destination, and a partner named PartnerTwo with a DUNS ID of 987654321 and an HTTP destination), this section describes how to add support for:

- The FTP transport
- Custom XML documents
- Binary files (with no packaging)

## Creating an FTP receiver

### About this task

The FTP receiver receives files and passes them to the Document Manager for processing. As described in "Configuring the FTP server for receiving documents" on page 33, before you can create an FTP receiver, you must have an FTP server installed, and you must have created an FTP directory and configured your FTP server.

In this example, it is assumed that the FTP server has been configured for Partner Two and that the root directory is c:/ftproot.

1. Click **Hub Admin > Hub Configuration > Receivers**.
2. Click **Create Receiver**.
3. Enter the following information:
   a. Receiver Name: **FTP_Receiver**
   b. Transport: **FTP Directory**
   c. FTP Root Directory: **C:/ftproot**
4. Click **Save**.

## Setting up the hub to receive binary files

This section covers the steps required to configure the hub to receive binary documents that Partner Two wants to send to the internal partner.

### Creating an interaction for binary documents

#### About this task

By default, WebSphere Partner Gateway provides four interactions involving binary documents. It does not, however, provide an interaction for binary documents packaged as None going to a partner with the document also packaged as None. In this section, you will create the required interaction to allow binary documents to pass through the system.

### Procedure

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Click **Create** from **Manage Interaction** view.
4. From **Source** select: **Package: None Protocol: Binary (1.0) Document Type: Binary (1.0)**.
5. From **Target** select: **Package: None Protocol: Binary (1.0) Document Type: Binary (1.0)**.
6. Optionally select **Transform** map.
7. From the **Action** list, select **Pass Through**.
8. Click **Save**.

## Updating the B2B capabilities for the internal partner

### About this task

This section shows how to configure the internal partner to be able to accept binary documents.

### Procedure

1. Click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Click the **View details** icon next to **Comm Man**.
4. Click **B2B Capabilities**.
5. Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.
6. Click the **Expand** icon next to **Package: None**.
7. Click the **Role is not active** icon for **Protocol: Binary (1.0)** under **Set Target**.
8. Click the **Expand** icon next to **Protocol: Binary (1.0)**.
9. Finally, click the **Role is not active** icon for **Document Type: Binary (1.0)** under **Set Target**.

## Updating the B2B capabilities for Partner Two

### About this task

This section shows how to configure Partner Two to be able to send binary documents.

### Procedure

1. Click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Click the **View details** icon next to Partner Two.
4. Click **B2B Capabilities**.
5. Click the **Role is not active** icon under **Set Source** for **Package: None** to enable it.
6. Click the **Expand** icon next to **Package: None**.
7. Click the **Role is not active** icon for **Protocol: Binary (1.0)** under **Set Source**.
8. Click the **Expand** icon next to **Protocol: Binary (1.0)**.
9. Finally, click the **Role is not active** icon for **Document Type: Binary (1.0)** under **Set Source**.

### Creating a new partner connection
#### About this task

This section shows how to configure a new partner connection between the internal partner and Partner Two for binary documents.

#### Procedure

1. Click **Account Admin > Connections**.
2. Select **Partner Two** from the **Source** list.
3. Select **Internal Partner** from the **Target** list.
4. Click **Search**.
5. Locate the **None (N/A), Binary (1.0), Binary (1.0)** to **None (N/A), Binary (1.0), Binary (1.0)** connection, and click **Activate** to activate it.

## Setting up the hub for custom XML documents

As described in "Custom XML document processing" on page 148, you must configure the hub to be able to route custom XML Files. This section covers the steps required to configure the Document Manager to be able to route the following XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Tester>
<Tester type="Test type A">
    <From>987654321</From>
    <To>123456789</To>
</Tester>
```

For this example, the Document Manager uses the RootTag to identify the type of XML document. It then extracts the values from the From and To fields to identify the From Partner business identifier and To Partner business identifier.

### Creating the CustomXML protocol definition format
#### About this task

The first step is to create a new protocol for the Custom XML you are going to exchange.

#### Procedure

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Create Document Definition**.
3. Select **Protocol** from the **Document Definition type** list.
4. Enter the following information:
   a. Code: **Custom XML**
   b. Version: **1.0**
   c. Description: **Example protocol definition**
5. Set **Document Level** to **No**.
6. Set **Status** to **Enabled**.
7. Set **Visibility: Hub Administrator** to **Yes**.
8. Set **Visibility: Internal Partner** to **Yes**.
9. Set **Visibility: Partner** to **Yes**.
10. Select:
    a. Package: **AS**

b. Package: **None**
c. Package: **Backend Integration**.

11. Click **Save**.

## Creating the Tester_XML document definition

### About this task

The second step is to create a document definition for the new protocol.

### Procedure

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Create Document Definition**.
3. Select **Document Type** from the **Document Definition type** list.
4. Enter the following information:
   a. Name: **Tester_XML**
   b. Version: **1.0**
   c. Description: **Example custom XML document type**
5. Set **Document Level** to **Yes**.
6. Set **Status** to **Enabled**.
7. Set **Visibility: Hub Administrator** to **Yes**.
8. Set **Visibility: Internal Partner** to **Yes**.
9. Set **Visibility: Partner** to **Yes**.
10. Click the **Expand** icon next to **Package: AS** and select **Protocol: CustomXML**.
11. Click the **Expand** icon next to **Package: None** and select **Protocol: CustomXML**.
12. Click the **Expand** icon next to **Package: Backend Integration** and select **Protocol: CustomXML**.
13. Click **Save**.

## Creating the Tester_XML Format

### About this task

Finally, you create the XML format associated with the new protocol.

### Procedure

1. Click **Hub Admin > Hub Configuration > XML Formats**.
2. Click **Create Document Family**.
3. Enter or select the following information:
   a. Family name: **Example family**
   b. Protocol: **Custom XML 1.0**
   c. Family type: **Root Tag**
   d. Large file option: **None**
   e. Family identifier: **Tester**
4. Click **Save**.
5. On the resulting Document family page, click **Create XML format**.
6. In the Document type list, select **Tester_XML**.
7. For the Format identifier value, enter **Test type A**.
8. For the XPath expression for the Format identifier, enter **/Tester/@type** .

9. Leave the Prefix Namespace field blank (no name spaces are used in the document), and the Return type as **Text**.
10. Enter **1** in both the Format version value field and the XPath expression field. Change the Return type to **Constant**. This means that all documents that have the Format identifier "Tester" will have the right version for a match with this format. This is because the version for all documents will be 1, and the version for this format is also 1. Therefore, the version always matches.
11. Enter **/Tester/From** for the XPath expression for the Source business identifier.
12. Enter **/Tester/To** for the XPath expression for the Target business identifier.
13. Leave the remaining fields in the format as they are. They are optional and are not used in this example.
14. Click **Save**.

## Creating an interaction for Tester_XML documents

### About this task

You now have a new protocol and document type with which to set up an interaction.

### Procedure

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interaction link**.
2. In the **Manage Interaction** screen, click **Create Interaction link**.
3. From **Source**, select:
   a. Package: **None**
   b. Protocol: **Custom XML (1.0)**
   c. Document Type: **Tester_XML (1.0)**.
4. From **Target** select:
   a. Package: **None**
   b. Protocol: **Custom XML (1.0)**
   c. Document Type: **Tester_XML (1.0)**.
5. From the **Action** list, select **Pass Through**.
6. Click **Save**.

## Updating the B2B capabilities for the internal partner

### About this task

To enable the exchange of the custom XML document, you must update the B2B capabilities of the partners.

First, enable the internal partner to receive (be the target for) Tester_XML documents.

### Procedure

1. Click **Account Admin > Profiles > Partner**.
2. Click **Search**.
3. Select the internal partner from the list of partners. (Note that this example assumes the internal partner has the business identifier 123456789.)
4. Click **B2B Capabilities**.
5. Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.

6. Click the **Expand** icon next to **Package: None**.
7. Click the **Role is not active** icon for **Protocol: Custom XML (1.0)** for **Set Target**.
8. Click the **Expand** icon next to **Protocol: Custom XML (1.0)**.
9. Finally, click the **Role is not active** icon for **Document Type: Tester_XML (1.0)** for **Set Target**.

## Updating the B2B capabilities for Partner Two

### About this task

You update the B2B capabilities of Partner Two to enable the exchange of messages using the new custom XML format .

Enable Partner Two to be the source of Tester_XML documents. (Note that the example assumes that Partner Two has a business identifier of 987654321.)

### Procedure

1. Click **Account Admin > Profiles> Partner**.
2. Click **Search**.
3. Select **Partner Two** from the list of partners. (Note that this example assumes Partner Two has the business identifier 987654321.)
4. Click **B2B Capabilities**.
5. Click the **Role is not active** icon under **Set Source** for **Package: None** to enable it.
6. Click the **Expand** icon next to **Package: None**.
7. Click the **Role is not active** icon for **Protocol: Custom XML (1.0)** for **Set Source**.
8. Click the **Expand** icon next to **Protocol: Custom XML (1.0)**.
9. Finally, click the **Role is not active** icon for **Document Type: Tester_XML (1.0)** for **Set Source**.

## Creating a new partner connection

### About this task

Finally, create a new partner connection.

### Procedure

1. Click **Account Admin > Connections.**
2. Select **Partner Two** from the **Source** list.
3. Select **Internal Partner** from the **Target** list.
4. Click **Search**.
5. Locate the **None (N/A), Custom XML (1.0), Tester_XML(1.0)** to **None (N/A), Custom XML(1.0), Tester_XML (1.0)** connection and click **Activate** to activate it.

## Routing a document using Custom XML

Copy the example XML from the beginning of this example and paste it into a text editor. Save the file on your machine with a name of your choice. Then send the file to the hub by dropping it into the directory used by the file receiver. Look in the document viewer and you should see that the document is routed from Partner Two to the internal partner using the connection that you defined for it.

# Chapter 20. EDI examples

This appendix provides examples of sending or receiving EDI interchanges and transforming them to and from XML and record-oriented data (ROD) documents.

The examples in this appendix are unrelated to those in Chapter 19, "Basic examples," on page 299. New targets, destinations, and profiles are created for the examples in this appendix.

**Note:** An example of an EDI interchange that is passed through the hub (no de-enveloping or transformation) is included in Chapter 19, "Basic examples," on page 299.

Each of these four examples is self-contained. For example, if you follow the EDI to XML example, you will see all the steps (from creating targets through activating connections) for that example.

This appendix includes the following topics:

- " EDI to ROD example"
- " EDI to XML example" on page 330
- " XML to EDI example" on page 335
- " ROD to EDI example" on page 342

These examples are intended to provide you with a quick overview of the steps required to configure a system. If you are using these examples to set up your system, modify the specific information (for example, names and business IDs) to suit the needs of your business.

## EDI to ROD example

This section provides an example of sending an EDI transaction (within an envelope) to the hub, where it is transformed into a record-oriented-data (ROD) document and sent to internal partner.

### De-enveloping and transforming an EDI interchange

#### About this task

In this example, it is assumed that the Data Interchange Services mapping specialist has created a transformation map that takes a standard EDI 850 transaction (defined with the X12V5R1 dictionary, corresponding to the Version 5010 of X12) and transforms it into a record-oriented document (ROD) that will be processed by the back-end application of the internal partner. In this example, the map is named S_DT_EDI_TO_ROD.eif.

The Data Interchange Services mapping specialist can export the transformation map directly to the WebSphere Partner Gateway database. Alternatively, the Data Interchange Services mapping specialist can send you the file, in which case you use the bcgDISImport utility to import it into WebSphere Partner Gateway. This appendix assumes the second scenario.

## Importing the transformation map
### About this task

This section describes the steps you take to import a transformation map that will take EDI input and transform it into record-oriented data (ROD) format. In the process of importing the transformation map, you also import the document definition associated with the map.

Before you can import the transformation map, the Data Interchange Services mapping specialist must send it to you. This set of steps assumes that the file, S_DT_EDI_TO_ROD.eif, is on your system.

1. Open a command window.
2. Enter the following command or script:
   - On a UNIX system:

     ```
     <ProductDir>/bin/bcgDISImport.sh <database_user_ID>
     <password> S_DT_EDI_TO_ROD.eif
     ```

   - On a Windows system:

     ```
     <ProductDir>\bin\bcgDISImport.bat <database_user_ID>
     <password> S_DT_EDI_TO_ROD.eif
     ```

     where *<database_user_ID>* and *<password>* are the values that you used when you installed the database as part of the WebSphere Partner Gateway installation.

## Verifying the transformation map and document definitions
### About this task

To verify that the transformation maps and document definitions you imported are available on the Community Console, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Maps > Transformation Maps**.

   The S_DT_EDI_TO_ROD map is displayed.
2. Click the **View details** icon next to the map.

   You see the document definitions with which this map is associated:

*Table 35. Document definition associated with the map*

| Source | Target |
|---|---|
| Package: N/A Protocol: X12V5R1 (ALL)Document Type: 850 (ALL) | Package: None Protocol: DEMO850CL_DICTIONARY(ALL) Document Type: DEMO850CLSUW (ALL) |

The S_DT_EDI_TO_ROD map was defined to take an X12 850 transaction (which adheres to the X12V5R1 standard) and transform it to a custom protocol (DEMO850CL_DICTIONARY) and document type (DEMO850CLSUW).

## Configuring the receiver
### About this task

In this section, you create a file-system directory receiver for the hub:

1. Click **Hub Admin > Hub Configuration > Receivers** and click **Create Receiver**.
2. For Receiver Name, type: **EDIFileTarget**.
3. From the Transport list, select **File Directory**.
4. For Root Path, type: **/Data/Manager/editarget**.

5. Click **Save**.

The partner sends the EDI interchange to this receiver.

## Creating the interactions
### About this task

You create two interactions--one for the EDI envelope and one for the transaction within the EDI envelope.

Create an interaction that represents the EDI envelope.

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Create Interaction**.
3. Under **Source**, expand **Package: None** and **Protocol: EDI-X12** and select **Document Type: ISA**.
4. Under **Target**, expand **Package: N/A** and **Protocol: EDI-X12** and select **Document Type: ISA.**
5. From the Action list, select **EDI De-envelope**.

   **Note:** No transformation is occurring in this interaction. The EDI interchange is being de-enveloped, resulting in the individual transaction (850). You do not, therefore, need a transformation map for this interaction.
6. Click **Save**.

Create an interaction that has a source that represents the 850 transaction and a target that represents the transformed document.

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click **Create Interaction**.
3. Under **Source**, expand **Package: N/A** and **Protocol: X12V5R1** and select **Document Type: 850**.
4. Under **Target**, expand **Package: None** and **Protocol: DEMO850CL_DICTIONARY** and select **Document Type: DEMO850CLSUW**.
5. From the Transformation Map list, select **S_DT_EDI_TO_ROD**.
6. From the Action list, select **EDI Validate and EDI Translate**.
7. Click **Save**.

This interaction represents the transformation of a standard EDI X12 850 transaction into a different format and, therefore, you must select a transformation map.

## Creating the partners
### About this task

For this example, you have two partners: the internal partner (Manager) and an external partner (TP1).

Create the Internal Partner profile:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type: **ComManager**
3. For Partner Display Name: type **Manager**
4. For Partner Type, select **Internal Partner**.
5. Click **New** for Business ID and type `000000000` as the Freeform ID.

**Note:** Make sure you select Freeform and not DUNS.

6. Click **New** again for Business ID and type `01-000000000` as the Freeform ID.
7. Click **Save**.

Create the second partner:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type **TP1**
3. For Partner Display Name, type **TP1**
4. For Partner Type, select **External Partner**.
5. Click **New** for Business ID and type `000000001` as the Freeform ID.

   **Note:** Make sure you select Freeform and not DUNS.
6. Click **New** again for Business ID and type `01-000000001` as the Freeform ID.
7. Click **Save**.

## Creating the destinations

### About this task

Create file-directory destinations for both partners in the example. First, create a destination for the Manager:

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon next to the Manager profile.
3. Click **Destinations** and then **Create**.
4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist on your file system.
   a. For Name, type **ManagerFileDestination**.
   b. From the Transport List, select **File Directory**.
   c. For Address, type: **file://Data/Manager/filedestination**
   d. Click **Save**.
5. Click **List** to list all the destinations for the internal partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4.
8. Click **Save**.

Next, create a destination for the partner.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Select the other partner you created for this example by clicking on the **View details** icon next to **TP1**.
3. Click **Destinations** and then **Create**.
4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist.
   a. For Name, type **TP1FileDestination**.
   b. From the Transport list, select **File Directory**.
   c. For Address, type: **file://Data/TP1/filedestination**
   d. Click **Save**.
5. Click **List** to list all the destinations for the partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4.

8. Click **Save**.

## Setting up B2B capabilities

### About this task

Enable the B2B capabilities of the two partners in this exchange. In this example, the EDI interchange is originating with an external partner (TP1) and will be delivered to the internal partner.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon for the source partner for this example (**TP1**).
3. Click **B2B Capabilities**.
4. Enable two sets of capabilities for the source partner.
   a. First, enable the document definition representing the EDI envelope:
      1) Click the **Role is not active** icon under **Set Source** for **Package: None** to enable it.
      2) Expand **Package: None**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol EDI-X12 (ALL)**.
      4) Expand **Protocol EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: ISA (ALL)**.
   b. Next, enable the document definition representing the 850 transaction:
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: X12V5R1 (ALL)**.
      4) Expand **Protocol: X12V5R1 (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: 850**.
5. Click **Account Admin > Profiles > Partner** and click **Search**.
6. Click the **View details** icon for the target partner for this example (**Manager**).
7. Click **B2B Capabilities**.
8. Enable two sets of capabilities for the target partner.
   a. First, enable the document definition representing the envelope:
      1) Click the **Role is not active** icon under **Set Target** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: EDI-X12 (ALL)**.
      4) Expand **Protocol: EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: ISA (ALL)**.
   b. Next, enable the document definition representing the transformed document:
      1) Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.
      2) Expand **Package: None**.

3) Click the **Role is not active** icon under **Set Target** for **Protocol: DEMO850CL_DICTIONARY (ALL)**.
4) Expand **Protocol: DEMO850CL_DICTIONARY (ALL)**.
5) Click the **Role is not active** icon under **Set Target** for **Document Type: DEMO850CLSUW(ALL)**.

### Activating the connections
#### About this task

To activate the connections:

1. Click **Account Admin > Connections**.
2. Select **TP1** from the Source list.
3. Select **Manager** from the Target list.
4. Click **Search**.
5. Click **Activate** for the connection that represents the envelope:

*Table 36. Envelope connection*

| Source | Target |
|---|---|
| Package: None (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) | Package: N/A (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA(ALL) |

6. Click **Activate** for the connection that represents the 850 transaction to the transformed document:

*Table 37. EDI transaction to ROD document connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: X12V5R1Document Type: 850 (ALL) | Package: None (N/A) Protocol: DEMO850CL_DICTIONARY (ALL) Document Type: DEMO850CLSUW (ALL) |

### Adding attributes
#### About this task

Set the attribute that allows documents with duplicate IDs:

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Click the **Expand** icon next to **Package: None**.
3. Click the **Edit Attribute Values** icon next to **Protocol: EDI-X12**.
4. Scroll down to the Document Type Context Attributes section of the page. In the **Allow documents with duplicate document ids** row, select **Yes** from the list.
5. Click **Save**.

At this point, if TP1 sent an EDI interchange containing an 850 transaction to the internal partner, the EDI interchange would be de-enveloped, resulting in an 850 transaction. The 850 transaction would then be transformed to the DEMO850CLSUW document type, and the transformed document would be sent to the destination of the internal partner.

## Adding a TA1 to the exchange

In X12, the TA1 is an optional segment that can be used to acknowledge receipt of an interchange. The sender can request a TA1 from the receiver by setting element

14 of the ISA Interchange Control Header to **1**. The Allow a TA1 request attribute in WebSphere Partner Gateway can be used to control whether a TA1 is sent when the sender requests it.

The &WDI_TA1_ACK map is installed during the installation of WebSphere Partner Gateway, so you do not have to import it.

## Creating the associations

### About this task

To associate the map with a document definition, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Maps > EDI FA Maps**.
   The &WDI_TA1_ACK map is displayed.
2. Click the **View details** icon next to the map.
   You see information about the map as well as a folder for each type of package available on the system.
3. Create the association to the document definition by performing these steps:
   a. Select the check box next to **Package: None** and expand the folder.
   b. Select the check box next to **Protocol: EDI-X12 (ALL)** and expand the folder.
   c. Select the check box next to **Document Type: ISA (ALL)**.
   d. Click **Save**.

You have created an association between the &WDI_TA1_ACK1 map and the document definition for the envelope.

## Creating interactions

### About this task

Create an interaction that represents the TA1 transaction.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Under **Source**, expand **Package: N/A** and **Protocol: &X44TA1** and select **Document Type: TA1**.
4. Under **Target**, expand **Package: N/A** and **Protocol: &X44TA1** and select **Document Type: TA1**.
5. From the Action list, select **Pass Through**.
6. Click **Save**.

Create an interaction that has a source that represents the EDI envelope that will hold the TA1.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Under **Source**, expand **Package: N/A** and **Protocol: EDI-X12** and select **Document Type: ISA**.
4. Under **Target**, expand **Package: None** and **Protocol: EDI-X12** and select **Document Type: ISA.**
5. From the Action list, select **Pass Through**.
6. Click **Save**.

## Enabling B2B capabilities

### About this task

Next, you add the newly created interactions to the B2B capabilities of the partners.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon for the source partner for this example (**Manager**).

   **Note:** Remember that the TA1 flows from the partner that receives the ROD document to the partner that sent it. In this example, the Manager is the source of the TA1 and partner TP1 is the target.
3. Click **B2B Capabilities**.
4. Enable two sets of capabilities for the source partner.
   a. First, enable the capability for the TA1.
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: &X44TA1**.
      4) Expand **Protocol: &X44TA1**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: TA1 (ALL)**.
   b. Next, enable the capability for the envelope:
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: EDI-X12**.
      4) Expand **Protocol: EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: ISA (ALL)**.
5. Click **Account Admin > Profiles > Partner** and click **Search**.
6. Click the **View details** icon for the target partner for this example (**TP1**).
7. Click **B2B Capabilities**.
8. Enable two sets of capabilities for the target partner.
   a. First, enable the document definition representing the TA1:
      1) Click the **Role is not active** icon under **Set Target** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: &X44TA1 (ALL)**.
      4) Expand **Protocol: &X44TA1 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: TA1 (ALL)**.
   b. Next, enable the document definition representing the EDI envelope:
      1) Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.
      2) Expand **Package: None**.

3) Click the **Role is not active** icon under **Set Target** for **Protocol: EDI-X12 (ALL)**.
4) Expand **Protocol: EDI-X12 (ALL)**.
5) Click the **Role is not active** icon under **Set Target** for **Document Type: ISA (ALL)**.

## Creating the envelope profile

### About this task

You next create the profile for the envelope that will contain the TA1:

1. Click **Hub Admin > Hub Configuration > EDI > Envelope Profile**.
2. Click **Create**.
3. Type the name of the profile: **EnvProf1**.
4. From the EDI Standard list, select **X12**.
5. The **General** button is selected by default. Type the following values for the general attributes of the envelope:
   - INTCTLLEN: **9**
   - GRPCTLLEN: **9**
   - TRXCTLLEN: **9**
   - MAXDOCS: **1000**
6. Click **Interchange**, and enter the following values for the interchange attributes:
   - ISA01: **01**
   - ISA02: **ISA0000002**
   - ISA03: **02**
   - ISA04: **ISA0000004**
   - ISA11: \
   - ISA12: **00501**
   - ISA15: **T**
7. Click **Save**.

## Activating partner connections

### About this task

To activate the connections:

1. Click **Account Admin > Connections**.
2. Select **Manager** from the Source list.
3. Select **TP1** from the Target list.
4. Click **Search**.
5. Activate the connection that represents the TA1.

*Table 38. TA1 connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: &X44TA1 (ALL) Document Type: TA1 (ALL) | Package: N/A (N/A) Protocol: &X44TA1 (ALL) Document Type: TA1 (ALL) |

6. Activate the connection that represents the envelope:

*Table 39. Envelope connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) | Package: None (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) |

### Configuring the attributes

#### About this task

To specify attributes for the envelope profile:

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Select **TP1** from the list.
3. Click **B2B Capabilities**.
4. Click the **Expand** icon next to **Package: None**.
5. Click the **Edit** icon next to **Protocol: EDI-X12 (ALL)**.
6. In the **Allow a TA1 Request** row, select **Yes**.
7. Click **Save**.
8. Click **B2B Capabilities** again.
9. Click the **Expand** icon next to **Package: N/A**.
10. Click the **Edit** icon next to **Protocol: &X44TA1 (ALL)**.
11. Specify the following attributes:
    a. In the Envelope Profile row, select **EnvProf1** from the list.
    b. In the Interchange qualifier row, type **01**.
    c. In the Interchange identifier row, type **000000001**.
    d. In the Interchange usage indicator row, type **T**.
12. Click **Save**.

In this series of tasks, you have added a TA1 acknowledgment to the exchange. When the interchange is received, WebSphere Partner Gateway sends a TA1 back to the sender (TP1). The TA1 is sent in an envelope that conforms to envelope profile EnvProf1.

## Adding an FA map

This section describes how to add a standard functional acknowledgment (997) to the flow described in " EDI to ROD example" on page 317. The functional acknowledgment provides confirmation to the sender that the transaction was received.

**Note:** This example is similar to " Adding a TA1 to the exchange" on page 322. However, it is not directly related to that example. Instead, it builds on the tasks you performed in " EDI to ROD example" on page 317.

WebSphere Partner Gateway includes a set of preinstalled functional acknowledgment map names that begin with $DT_FA. This is followed by the name of the functional acknowledgment message and the version and release of the message. For example, Version 2 Release 4 of the 997 functional acknowledgment message is named $DT_997V2R4. See "Setting up acknowledgments" on page 203 for the list of maps provided with WebSphere Partner Gateway.

**Related information**

## Creating the associations
### About this task

To associate the map with a document definition, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Maps > EDI FA Maps**.
   The &DT_FA997V2R4 map is displayed.
2. Click the **View details** icon next to the map.
   You see information about the map as well as a folder for each type of package available on the system.
3. Create the association to the document definition by performing these steps:
   a. Select the check box next to **Package: N/A** and expand the folder
   b. Select the check box next to **Protocol: X12V5R1** and expand the folder.
   c. Select the check box next to **Document Type: 850**.
   d. Click **Save**.

You have associated this functional acknowledgment 997 map with the X12 protocol.

## Creating interactions
### About this task

Create an interaction that represents the 997 acknowledgment.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Under **Source**, expand **Package: N/A** and **Protocol: &DT99724** and select **Document Type: 997**.
4. Under **Target**, expand **Package: N/A** and **Protocol: &DT99724** and select **Document Type: 997**.
5. From the Action list, select **Pass Through**.
6. Click **Save**.

Create an interaction that represents the envelope.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand **Package: N/A** and **Protocol: EDI-X12** and select **Document Type: ISA**.
4. Expand **Package: None** and **Protocol: EDI-X12** and select **Document Type: ISA**.
5. From the Action list, select **Pass Through**.
6. Click **Save**.

## Enabling B2B capabilities
### About this task

Next, you add the newly created interactions to the B2B capabilities of the partners.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon for the source partner for this example (**Manager**).

**Note:** Remember that the functional acknowledgment flows from the partner that receives the ROD document to the partner that sent it. In this example, the Manager is the source of the functional acknowledgment, and partner TP1 is the target.

3. Click **B2B Capabilities**.
4. Enable two sets of capabilities for the source partner.
   a. First, enable the capability for the FA.
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: &DT99724**.
      4) Expand **Protocol: &DT99724**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: 997 (ALL)**.
   b. Next, enable the capability for the envelope:
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: EDI-X12**.
      4) Expand **Protocol: EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: ISA (ALL)**.
5. Click **Account Admin > Profiles > Partner** and click **Search**.
6. Click the **View details** icon for the target partner for this example (**TP1**).
7. Click **B2B Capabilities**.
8. Enable two sets of capabilities for the target partner.
   a. First, enable the document definition representing the 997:
      1) Click the **Role is not active** icon under **Set Target** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: &DT99724 (ALL)**.
      4) Expand **Protocol: &DT99724 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: 997 (ALL)**.
   b. Next, enable the document definition representing the EDI envelope:
      1) Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.
      2) Expand **Package: None**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: EDI-X12 (ALL)**.
      4) Expand **Protocol: EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: ISA(ALL)**.

## Creating the envelope profile

### About this task

You next create the profile for the envelope that will contain the 997 functional acknowledgment. A functional acknowledgment, like a transaction, must be enveloped before it can be sent.

1. Click **Hub Admin > Hub Configuration > EDI > Envelope Profile**.
2. Click **Create**.
3. Type the name of the profile: **EnvProf1**.
4. From the EDI Standard list, select **X12**.
5. The **General** button is selected by default. Type the following values for the general attributes of the envelope:
   - INTCTLLEN: **9**
   - GRPCTLLEN: **9**
   - TRXCTLLEN: **9**
   - MAXDOCS: **1000**
6. Click the **Interchange** button and type the following values for the interchange attributes:
   - ISA01: **01**
   - ISA02: **ISA0000002**
   - ISA03: **02**
   - ISA04: **ISA0000004**
   - ISA11: \
   - ISA12: **00501**
   - ISA15: **T**
7. Click **Save**.

## Activating partner connections

### About this task

To activate the connections:

1. Click **Account Admin > Connections**.
2. Select **Manager** from the Source list.
3. Select **TP1** from the Target list.
4. Click **Search**.
5. Click **Activate** for the connection that represents the 997 functional acknowledgment:

*Table 40. Functional acknowledgment connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: &DT99724 (ALL) Document Type: 997 (ALL) | Package: N/A (N/A) Protocol: &DT99724 (ALL) Document Type: 997 (ALL) |

6. Click **Activate** for the connection that represents the EDI envelope being sent back to the originator of the exchange.

*Table 41. Envelope connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) | Package: None (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) |

### Configuring attributes

#### About this task

First, you specify which FA map to use:

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Select **TP1** from the list.
3. Click **B2B Capabilities**.
4. Click the **Expand** icon next to **Package: N/A**.
5. Click the **Edit** icon next to **Protocol: X12V5R1 (ALL)**.
6. In the FA Map row, select **&DT_FA997V2R4**.
7. Click **B2B Capabilities** again.
8. Click the **Expand** icon next to **Package: N/A**.
9. Click the **Edit** icon next to **Protocol: &DT99724 (ALL)**.
10. Specify the following attributes:
    a. In the Envelope Profile row, select **EnvProf1** from the list.
    b. In the Interchange qualifier row, type **01**.
    c. In the Interchange identifier row, type **000000001**.
    d. In the Interchange usage indicator row, type **T**.
11. Click **Save**.

In this series of tasks, you have added an EDI-X12 997 functional acknowledgment to the exchange, so that when the internal partner receives the document, it sends the 997 back to the sender (TP1). The 997 acknowledgment is sent in an envelope that conforms to envelope profile EnvProf1.

## EDI to XML example

This section provides an example of sending an EDI transaction (within an envelope) to the hub, where it is transformed into an XML document and sent to the internal partner.

In this example, it is assumed that the Data Interchange Services mapping specialist has created a transformation map that takes a standard EDI 879 transaction (defined with the X12V5R1 dictionary, corresponding to the Version 5010 of X12) and transforms it into an XML document that will be processed by the back-end application of the internal partner. In this example, the map is named S_DT_EDI_TO_XML.eif.

The Data Interchange Services mapping specialist can export the transformation map directly to the WebSphere Partner Gateway database. Alternatively, the Data Interchange Services mapping specialist can send you the file, in which case you use the bcgDISImport utility to import it into WebSphere Partner Gateway. This appendix assumes the second scenario.

## Importing the transformation map

### About this task

This section describes the steps you take to import a transformation map that will take EDI input and transform it into XML format. In the process of importing the transformation map, you also import the document definition associated with the map.

Before you can import the transformation map, the Data Interchange Services mapping specialist must send it to you. This set of steps assumes that the file, S_DT_EDI_TO_XML.eif, is on your system.

1. Open a command window.
2. Enter the following command or script:
   - On a UNIX system:

     ```
     <ProductDir>/bin/bcgDISImport.sh <database_user_ID>
      <password> S_DT_EDI_TO_XML.eif
     ```

   - On a Windows system:

     ```
     <ProductDir>\bin\bcgDISImport.bat <database_user_ID>
      <password> S_DT_EDI_TO_XML.eif
     ```

     where *<database_user_ID>* and *<password>* are the values that you used when you installed the database as part of the WebSphere Partner Gateway installation.

## Verifying the transformation map and document definitions

### About this task

To verify that the transformation maps and document definitions you imported are available on the Community Console, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Maps > Transformation Maps**.

   The S_DT_EDI_TO_XML map is displayed.
2. Click the **View details** icon next to the map.

   You see the document definitions with which this map is associated:

*Table 42. Document definition associated with the map*

| Source | Target |
|---|---|
| Package: N/A Protocol: X12V5R1Document Type: 879 (ALL) | Package: None Protocol: FVT-XML-TEST (ALL) Document Type: WWRE_ITEMCREATIONINTERNAL (ALL) |

The S_DT_EDI_TO_XML map was defined to take an X12 879 transaction (which adheres to the X12V5R1 standard) and transform it to a custom protocol.

## Configuring the receiver

### About this task

In this section, you create a file-system directory receiver for the hub:

1. Click **Hub Admin > Hub Configuration > Receivers** and click **Create Receiver**.
2. For Receiver Name, type: **EDIFileTarget**.
3. From the Transport list, select **File Directory**.

4. For Root Path, type: **/Data/Manager/editarget**.
5. Click **Save**.

The partner sends the EDI interchange to this receiver.

## Creating the interactions

### About this task

You create two interactions--one for the EDI envelope and one for the transaction within the EDI envelope.

Create an interaction that represents the EDI envelope.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand **Package: None** and **Protocol: EDI-X12** and select **Document Type: ISA**.
4. Expand **Package: N/A** and **Protocol: EDI-X12** and select **Document Type: ISA.**
5. From the Action list, select **EDI De-envelope**.

   **Note:** No transformation is occurring in this interaction. The EDI interchange is being de-enveloped, resulting in the individual transaction (879). You do not, therefore, need a transformation map for this interaction.
6. Click **Save**.

Create an interaction that has a source that represents the 879 transaction and a target the represents the transformed document.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand **Package: N/A** and **Protocol: X12V5R1** and select **Document Type: 879**.
4. Expand **Package: None** and **Protocol: FVT-XML-TEST** and select **Document Type: WWRE_ITEMCREATIONINTERNAL**.
5. From the Transformation Map list, select **S_DT_EDI_TO_XML**.
6. From the Action list, select **EDI Validate and EDI Translate**.
7. Click **Save**.

This interaction represents the transformation of a standard EDI X12 879 transaction into a different format and, therefore, you must select a transformation map.

## Creating the partners

### About this task

For this example, you have two partners: the internal partner (Manager) and an external partner (TP1).

Create the Internal Partner profile:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type: **ComManager**
3. For Partner Display Name: type **Manager**

4. For Partner Type, select **Internal Partner**.
5. Click **New** for Business ID and type `000000000` as the Freeform ID.

   **Note:** Make sure you select Freeform and not DUNS.
6. Click **New again** for Business ID and type `01-000000000` as the Freeform ID.
7. Click **Save**.

Create the second partner:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type **TP1**
3. For Partner Display Name, type **TP1**
4. For Partner Type, select **External Partner**.
5. Click **New** for Business ID and type `000000001` as the Freeform ID.

   **Note:** Make sure you select Freeform and not DUNS.
6. Click **New** again for Business ID and type `01-000000001` as the Freeform ID.
7. Click **Save**.

## Creating the destinations

### About this task

Create file-directory destinations for both partners in the example. First, create a destination for the Manager:

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon next to the Manager profile.
3. Click **Destinations** and then **Create**.
4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist on your file system.
   a. For Name, type **ManagerFileDestination**.
   b. From the Transport List, select **File Directory**.
   c. For Address, type: **file://Data/Manager/filedestination**
   d. Click **Save**.
5. Click **List** to list all the destinations for the internal partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4.
8. Click **Save**.

Next, create a destination for the partner.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Select the other partner you created for this example by clicking on the **View details** icon next to **TP1**.
3. Click **Destinations** and then **Create**.
4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist.
   a. For Name, type **TP1FileDestination**.
   b. From the Transport list, select **File Directory**.
   c. For Address, type: **file://Data/TP1/filedestination**
   d. Click **Save**.

5. Click **List** to list all the destinations for the partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4 on page 333.
8. Click **Save**.

## Setting up B2B capabilities

### About this task

Enable the B2B capabilities of the two partners in this exchange. In this example, the EDI interchange is originating with an external partner (TP1) and will be delivered to the internal partner.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon for the source partner for this example (**TP1**).
3. Click **B2B Capabilities**.
4. Enable two sets of capabilities for the source partner.
   a. First, enable the document definition representing the EDI envelope:
      1) Click the **Role is not active** icon under **Set Source** for **Package: None** to enable it.
      2) Expand **Package: None**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol EDI-X12 (ALL)**.
      4) Expand **Protocol EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: ISA (ALL)**.
   b. Next, enable the document definition representing the transaction:
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: X12V5R1 (ALL)**.
      4) Expand **Protocol: X12V5R1 (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: 879**.
5. Click **Account Admin > Profiles > Partner** and click **Search**.
6. Click the **View details** icon for the target partner for this example (**Manager**).
7. Click **B2B Capabilities**.
8. Enable two sets of capabilities for the target partner.
   a. First, enable the document definition:
      1) Click the **Role is not active** icon under **Set Target** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: EDI-X12 (ALL)**.
      4) Expand **Protocol: EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: ISA (ALL)**.

b. Next, enable the document definition representing the transformed document:
   1) Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.
   2) Expand **Package: None**.
   3) Click the **Role is not active** icon under **Set Target** for **Protocol: FVT-XML-TEST (ALL)**.
   4) Expand **Protocol: FVT-XML-TEST (ALL)**.
   5) Click the **Role is not active** icon under **Set Target** for **Document Type: WWRE_ITEMCREATIONINTERNAL(ALL)**.

## Activating the connections

### About this task

To activate the connections:

1. Click **Account Admin > Connections**.
2. Select **TP1** from the Source list.
3. Select **Manager** from the Target list.
4. Click **Search**.
5. Click **Activate** for the connection that represents the envelope:

*Table 43. Envelope connection*

| Source | Target |
|---|---|
| Package: None (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) | Package: N/A (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) |

6. Click **Activate** for the connection that represents the 879 transaction to the transformed document:

*Table 44. EDI transaction to XML document connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: X12V5R1 (ALL)Document Type: 879 (ALL) | Package: None (N/A) Protocol: FVT-XML-TEST (ALL) Document Type: WWRE_ITEMCREATIONINTERNAL (ALL) |

At this point, if TP1 sent an EDI interchange containing an 879 transaction to the internal partner, the EDI interchange would be de-enveloped, resulting in an 879 transaction. The 879 transaction would then be transformed and the transformed document would be sent to the destination of the internal partner.

# XML to EDI example

This section provides an example of the internal partner sending an XML document to the hub, where it is transformed into an EDI transaction, enveloped within an EDI interchange, and sent to a partner.

In this example, it is assumed that the Data Interchange Services mapping specialist has created a transformation map that takes an XML document and transforms it into a standard EDI 850 transaction (defined with the MX12V3R1 dictionary) that will be processed by the partner. In this example, the map is named S_DT_XML_TO_EDI.eif.

The Data Interchange Services mapping specialist can export the transformation map directly to the WebSphere Partner Gateway database. Alternatively, the Data Interchange Services mapping specialist can send you the file, in which case you use the bcgDISImport utility to import it into WebSphere Partner Gateway. This appendix assumes the second scenario.

## Importing the transformation map

### About this task

This section describes the steps you take to import a transformation map that will take XML input and transform it into an EDI transaction. In the process of importing the transformation map, you also import the document definition associated with the map.

Before you can import the transformation map, the Data Interchange Services mapping specialist must send it to you. This set of steps assumes that the file, S_DT_XML_TO_EDI.eif, is on your system.

1. Open a command window.
2. Enter the following command or script:
   - On a UNIX system:

     ```
     <ProductDir>/bin/bcgDISImport.sh <database_user_ID>
      <password> S_DT_XML_TO_EDI.eif
     ```

   - On a Windows system:

     ```
     <ProductDir>\bin\bcgDISImport.bat <database_user_ID>
      <password> S_DT_XML_TO_EDI.eif
     ```

     where *<database_user_ID>* and *<password>* are the values that you used when you installed the database as part of the WebSphere Partner Gateway installation.

## Verifying the transformation map and document definitions

### About this task

To verify that the transformation maps and document definitions you imported are available on the Community Console, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Maps > Transformation Maps**.
   The S_DT_XML_TO_EDI map is displayed.
2. Click the **View details** icon next to the map.
   You see the document definitions with which this map is associated:

*Table 45. Document definitions associated with the map*

| Source | Target |
|---|---|
| Package: None Protocol: FVT-XML-TEST (ALL) Document Type: ICGCPO (ALL) | Package: N/A Protocol: MX12V3R1 (ALL) Document Type: 850 (ALL) |

The S_DT_XML_TO_EDI map was defined to take an XML document and transform it to an EDI transaction.

## Configuring the receiver

### About this task

In this section, you create a file-system directory receiver for the hub:

1. Click **Hub Admin > Hub Configuration > Receivers** and click **Create Receiver**.
2. For Receiver Name, type: **XMLFileTarget**.
3. From the Transport list, select **File Directory**.
4. For Root Path, type: **/Data/Manager/xmltarget**.
5. From the Configuration Point list, select **Preprocess**.
6. Select **com.ibm.bcg.edi.receiver.preprocesshandler.XMLSplitterHandler** from the Available List and click **Add** to move it to the Configured List.
7. Click **Save**.

The internal partner sends the XML document to this receiver.

## Creating the interactions

### About this task

You create two interactions--one for the XML-to-EDI transformation and one for the EDI envelope.

Create an interaction that has a source that represents the XML document and a target that represents the transformed 850 transaction.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand **Package: None** and **Protocol: FVT-XML-TEST** and select **Document Type: ICGCPO**.
4. Expand **Package: N/A** and **Protocol: MX12V3R1** and select **Document Type: 850**.
5. From the Transformation Map list, select **S_DT_XML_TO_EDI**.
6. From the Action list, select **XML Translate and EDI Validate**.
7. Click **Save**.

This interaction represents the transformation of an XML document into an EDI transaction and, therefore, you must select a transformation map.

Create an interaction that represents the EDI envelope.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand **Package: N/A** and **Protocol: EDI-X12** and select **Document Type: ISA**.
4. Expand **Package: None** and **Protocol: EDI-X12** and select **Document Type: ISA.**
5. From the Action list, select **Pass Through**.

   **Note:** No transformation is occurring in this interaction.
6. Click **Save**.

## Creating the partners

### About this task

For this example, you have two partners: the internal partner (Manager) and an external partner (TP1).

Create the Internal Partner profile:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type: **ComManager**
3. For Partner Display Name, type: **Manager**.
4. For Partner Type, select **Internal Partner**.
5. Click **New** for Business ID and type `000000000` as the Freeform ID.

   **Note:** Make sure you select Freeform and not DUNS.
6. Click **New** to create a new Business ID again and type `01-000000000` as the FreeForm ID. When you click New, the Email ID text box also enables and displays for you to create an Email ID.
7. Click **New** to create a new Email ID and type your Email ID in Email Identifier. Similarly, you can click New to create multiple Email IDs.
8. Click **Save**.

Create the second partner:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type **TP1**
3. For Partner Display Name, type **TP1**
4. For Partner Type, select **External Partner**.
5. Click **New** to create a new Business ID again and type `01-000000000` as the FreeForm ID. When you click New, the Email ID text box also enables and displays for you to create an Email ID.

   **Note:** Make sure you select Freeform and not DUNS.
6. Click **New** to create a new Email ID and type your Email ID in Email Identifier. Similarly, you can click New to create multiple Email IDs.
7. Click **Save**.

## Creating the destinations

### About this task

Create file-directory destinations for both partners in the example. First, create a destination for the Manager:

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon next to the Manager profile.
3. Click **Destinations** and then **Create**.
4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist on your file system.
   a. For Name, type **ManagerFileDestination**.
   b. From the Transport List, select **File Directory**.
   c. For Address, type: **file://Data/Manager/filedestination**
   d. Click **Save**.
5. Click **List** to list all the destinations for the internal partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4.
8. Click **Save**.

Next, create a destination for the partner.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Select the other partner you created for this example by clicking on the **View details** icon next to **TP1**.
3. Click **Destinations** and then **Create**.
4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist.
   a. For Name, type **TP1FileDestination**.
   b. From the Transport list, select **File Directory**.
   c. For Address, type: **file://Data/TP1/filedestination**
   d. Click **Save**.
5. Click **List** to list all the destinations for the partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4.
8. Click **Save**.

## Setting up B2B capabilities

### About this task

Enable the B2B capabilities of the two partners in this exchange. In this example, the XML document is originating from the internal partner and will be delivered to the external partner.

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon for the source partner for this example (**ComMan**).
3. Click **B2B Capabilities**.
4. Enable three sets of capabilities for the source partner.
   a. Enable the document definition representing the XML document:
      1) Click the **Role is not active** icon under **Set Source** for **Package: None** to enable it.
      2) Expand **Package: None**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: FVT-XML-TEST (ALL)**.
      4) Expand **Protocol: FVT-XML-TEST (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: ICGCPO (ALL)**.
   b. Next, enable the document definition representing the transformed document:
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: MX12V3R1(ALL)**.
      4) Expand **Protocol: MX12V3R1 (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: 850**.
   c. Then, enable the document definition representing the EDI envelope:
      1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.

3) Click the **Role is not active** icon under **Set Source** for **Protocol EDI-X12 (ALL)**.
4) Expand **Protocol EDI-X12 (ALL)**.
5) Click the **Role is not active** icon under **Set Source** for **Document Type: ISA (ALL)**.

5. Click **Account Admin > Profiles > Partner** and click **Search**.
6. Click the **View details** icon for the target partner for this example (**TP1**).
7. Click **B2B Capabilities**.
8. Enable two sets of capabilities for the target partner.
   a. First, enable the document definition representing the EDI 850 transaction:
      1) Click the **Role is not active** icon under **Set Target** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: MX12V3R1 (ALL)**.
      4) Expand **Protocol: MX12V3R1 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: 850 (ALL)**.
   b. Next, enable the document definition:
      1) Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.
      2) Expand **Package: None**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: EDI-X12 (ALL)**.
      4) Expand **Protocol: EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: ISA(ALL)**.

## Creating the envelope profile

### About this task

You next create the profile for the envelope that will contain the transformed 850 transaction.

1. Click **Hub Admin > Hub Configuration > EDI > Envelope Profile**.
2. Click **Create**.
3. Type the name of the profile: **EnvProf1**.
4. From the EDI Standard list, select **X12**.
5. The **General** button is selected by default. Type the following values for the general attributes of the envelope:
   - INTCTLLEN: **9**
   - GRPCTLLEN: **9**
   - TRXCTLLEN: **9**
   - MAXDOCS: **1000**
6. Click **Interchange**, and enter the following values for the interchange attributes:
   - ISA01: **01**
   - ISA02: **ISA0000002**
   - ISA03: **02**

- ISA04: **ISA0000004**
- ISA11: **U**
- ISA12: **00301**
- ISA15: **T**

7. Click **Save**.

# Creating the XML format

## About this task

In this section, you create the custom XML format.

1. Click **Hub Admin > Hub Configuration > XML Formats**.
2. Click **Create XML Format**.
3. For Routing Format, select **FVT-XML-TEST ALL**.
4. For File Type, select **XML**.
5. For Identifier Type, select **Root Tag** and type **MMDoc**.
6. For Source Business Id, select **Constant** and type **000000000**.
7. For Target BusinessId, select **Constant** and type **000000001**.
8. For Source Document Type, select **Constant** and type **ICGCPO**.
9. For Source Document Type Version, select **Constant** and type **ALL**.
10. Click **Save**.

# Activating the connections

## About this task

Activate the partner connections:

1. Click **Account Admin > Connections**.
2. Select **Manager** from the Source list.
3. Select **TP1** from the Target list.
4. Click **Search**.
5. Click **Activate** for the following connection:

*Table 46. XML document to EDI transaction connection*

| Source | Target |
|---|---|
| Package: None (N/A) Protocol: FVT-XML-TEST (ALL) Document Type: ICGCPO (ALL) | Package: N/A (N/A) Protocol: MX12V3R1 (ALL) Document Type: 850 (ALL) |

6. Click **Activate** for the connection that represents the EDI envelope:

*Table 47. EDI envelope connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) | Package: None (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) |

## Configuring attributes

### About this task

Configure the B2B Capabilities attributes of the target partner (TP1) and the source partner (Manager):

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon next to **TPI** to select it.
3. Click **B2B Capabilities**.
4. Click the **Expand** icon next to **Package: N/A**.
5. Click the **Edit** icon next to **Protocol: MX12V3R1**.
6. Specify the following attributes:
   a. In the Envelope Profile row, select **EnvProf1** from the list.
   b. In the Interchange qualifier row, type **01**.
   c. In the Interchange identifier row, type **000000001**.
   d. In the Interchange usage indicator row, type **T**.
7. Click **Save**.
8. Click **Account Admin > Profiles > Partner** and click **Search**.
9. Click the View details next to **Manager** to select it.
10. Click **B2B Capabilities**.
11. Click the **Expand** icon next to **Package: N/A**.
12. Click the **Edit** icon next to **Protocol: MX12V3R1 (ALL)**.
13. Specify the following attributes:
    a. In the Interchange qualifier row, type **01**.
    b. In the Interchange identifier row, type **000000000**.
    c. In the Interchange usage indicator row, type **T**.
14. Click **Save**.

At this point, if the source partner (the internal partner) sent an XML document to the partner, it would be translated (at the hub) to an EDI transaction, enveloped, and then sent to the partner's destination.

# ROD to EDI example

This section provides an example of the internal partner sending a ROD document to the hub, where it is transformed into an EDI transaction, enveloped within an EDI interchange, and sent to a partner.

In this example, it is assumed that the Data Interchange Services mapping specialist has created a transformation map that takes a record-oriented document (ROD) and transforms it into a standard EDI 850 transaction (defined with the X12V5R1 dictionary, corresponding to the Version 5010 of X12) that will be processed by the partner. In this example, the map is named S_DT_ROD_TO_EDI.eif.

The Data Interchange Services mapping specialist can export the transformation map directly to the WebSphere Partner Gateway database. Alternatively, the Data Interchange Services mapping specialist can send you the file, in which case you use the bcgDISImport utility to import it into WebSphere Partner Gateway. This appendix assumes the second scenario.

# Importing the transformation map

## About this task

This section describes the steps you take to import a transformation map that will take ROD input and transform it into an X12 transaction. In the process of importing the transformation map, you also import the document definition associated with the map.

Before you can import the transformation map, the Data Interchange Services mapping specialist must send it to you. This set of steps assumes that the file, S_DT_ROD_TO_EDI.eif, is on your system.

1. Open a command window.
2. Enter the following command or script:
   - On a UNIX system:

     ```
     <ProductDir>/bin/bcgDISImport.sh <database_user_ID>
      <password> S_DT_ROD_TO_EDI.eif
     ```

   - On a Windows system:

     ```
     <ProductDir>\bin\bcgDISImport.bat <database_user_ID>
      <password> S_DT_ROD_TO_EDI.eif
     ```

     where <*database_user_ID*> and <*password*> are the values that you used when you installed the database as part of the WebSphere Partner Gateway installation.

# Verifying the transformation map and document definitions

## About this task

To verify that the transformation maps and document definitions you imported are available on the Community Console, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Maps > Transformation Maps**.

   The S_DT_ROD_TO_EDI map is displayed.
2. Click the **View details** icon next to the map.

   You see the document definitions with which this map is associated:

*Table 48. Document definitions associated with the map*

| Source | Target |
|---|---|
| Package: None Protocol: ROD-TO-EDI_DICT (ALL) Document Type: DTROD-TO-EDI_ROD (ALL) | Package: N/A Protocol: X12V5R1(ALL) Document Type: 850 (ALL) |

The S_DT_ROD_TO_EDI map was defined to take a ROD document associated with the ROD-TO-EDI_DICT dictionary and transform it to an X12 850 transaction that conforms to the X12V5R1 standard.

# Configuring the receiver

## About this task

In this section, you create a file-system directory receiver for the hub:

1. Click **Hub Admin > Hub Configuration > Receivers** and click **Create Receiver**.
2. For Receiver Name, type: **RODFileTarget**.

3. From the Transport list, select **File Directory**.
4. For Root Path, type: **/Data/Manager/rodtarget**.
5. From the Configuration Point list, select **Preprocess**.
6. Select **com.ibm.bcg.edi.receiver.preprocesshandler.RODSplitterHandler** from the Available List and click **Add** to move it to the Configured List.
7. Select **com.ibm.bcg.edi.receiver.preprocesshandler.RODSplitterHandler** from the Configured List and click **Configure**.
8. Add the values shown in table:

*Table 49. ROD Splitter Handler attributes*

| Field | Value |
|---|---|
| From Packaging Name | None |
| From Packaging Version | N/A |
| From Protocol Name | ROD-TO-EDI_DICT |
| From Protocol Version | ALL |
| From Process Code | DTROD-TO-EDI_ROD |
| From Process Version | ALL |
| METADICTIONARY | ROD-TO-EDI_DICT |
| METADOCUMENT | DTROD-TO-EDI_ROD |
| METASYNTAX | rod |
| ENCODING | ascii |
| BCG_BATCHDOCS | ON |

9. Click **Set Values**.
10. Click **Save**.

The internal partner sends the ROD document to this target.

# Creating the interactions

## About this task

You create two interactions--one for the EDI envelope that will be sent from the hub and one for the transformation of the ROD document to EDI.

Create an interaction that has a source that represents the ROD document and a target the represents the X12 document.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand **Package: None** and **Protocol: ROD-TO-EDI_DICT** and select **DTROD-TO-EDI_ROD**.
4. Expand **Package: N/A** and **Protocol: X12V5R1** and select **Document Type: 850**.
5. From the Transformation Map list, select **S_DT_ROD_TO_EDI**.
6. From the Action list, select **ROD Translate and EDI Validate**.
7. Click **Save**.

This interaction represents the transformation of a ROD document into a standard X12 transaction and, therefore, you must select a transformation map.

Create an interaction that represents the EDI envelope.

1. Click **Hub Admin > Hub Configuration > Document Definition > Manage Interactions link**.
2. In the **Manage Interactions** screen, click **Create Interaction link**.
3. Expand **Package: N/A** and **Protocol: EDI-X12** and select **Document Type: ISA**.
4. Expand **Package: None** and **Protocol: EDI-X12** and select **Document Type: ISA.**
5. From the Action list, select **Pass Through**.

   **Note:** No transformation is occurring in this interaction. This interaction is to envelope the EDI interchange.
6. Click **Save**.

## Creating the partners

### About this task

For this example, you have two partners: the internal partner (Manager) and an external partner (TP1).

Create the Internal Partner profile:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type: **ComManager**
3. For Partner Display Name: type **Manager**
4. For Partner Type, select **Internal Partner**.
5. Click **New** for Business ID and type `000000000` as the Freeform ID.

   **Note:** Make sure you select Freeform and not DUNS.
6. Click **New** again for Business ID and type `01-000000000` as the Freeform ID.
7. Click **Save**.

Create the second partner:

1. Click **Account Admin > Profiles > Partner** and click **Create**.
2. For Company Login Name, type **TP1**
3. For Partner Display Name, type **TP1**
4. For Partner Type, select **External Partner**.
5. Click **New** for Business ID and type `000000001` as the Freeform ID.

   **Note:** Make sure you select Freeform and not DUNS.
6. Click **New** again for Business ID and type `01-000000001` as the Freeform ID.
7. Click **Save**.

## Creating the destinations

### About this task

Create file-directory destinations for both partners in the example. First, create a destination for the Manager:

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon next to the Manager profile.
3. Click **Destinations** and then **Create**.

4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist on your file system.
   a. For Name, type **ManagerFileDestination**.
   b. From the Transport List, select **File Directory**.
   c. For Address, type: **file://Data/Manager/filedestination**
   d. Click **Save**.
5. Click **List** to list all the destinations for the internal partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4
8. Click **Save**.

Next, create a destination for the partner.

1. Click **Account Admin> Profiles > Partner** and click **Search**.
2. Select the other partner you created for this example by clicking on the **View details** icon next to **TP1**.
3. Click **Destinations** and then **Create**.
4. Enter the following values for the destination. Remember that the file directory (the entire path) must already exist.
   a. For Name, type **TP1FileDestination**.
   b. From the Transport list, select **File Directory**.
   c. For Address, type: **file://Data/TP1/filedestination**
   d. Click **Save**.
5. Click **List** to list all the destinations for the partner.
6. Click **View Default Destinations**.
7. From the **Production** list, select the destination you created in step 4.
8. Click **Save**.

## Setting up B2B capabilities

### About this task

Enable the B2B capabilities of the two partners in this exchange. In this example, the ROD document is originating from the internal partner and will be delivered to the external partner (TP1).

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Click the **View details** icon for the source partner for this example (**Manager**).
3. Click **B2B Capabilities**.
4. Enable two sets of capabilities for the source partner.
   a. First, enable the document definition representing the ROD document:
      1) Click the **Role is not active** icon under **Set Source** for **Package: None** to enable it.
      2) Expand **Package: None**.
      3) Click the **Role is not active** icon under **Set Source** for **Protocol: ROD-TO-EDI_DICT (ALL)**.
      4) Expand **Protocol: ROD-TO-EDI_DICT (ALL)**.
      5) Click the **Role is not active** icon under **Set Source** for **Document Type: DTROD-TO-EDI_ROD (ALL)**.
   b. Next, enable the document definition representing the EDI envelope:

1) Click the **Role is not active** icon under **Set Source** for **Package: N/A** to enable it.
2) Expand **Package: N/A**.
3) Click the **Role is not active** icon under **Set Source** for **Protocol EDI-X12 (ALL)**.
4) Expand **Protocol EDI-X12 (ALL)**.
5) Click the **Role is not active** icon under **Set Source** for **Document Type: ISA (ALL)**.

5. Click **Account Admin > Profiles > Partner** and click **Search**.
6. Click the **View details** icon for the target partner for this example (**TP1**).
7. Click **B2B Capabilities**.
8. Enable two sets of capabilities for the target partner.
   a. First, enable the document definition representing the EDI 850 transaction:
      1) Click the **Role is not active** icon under **Set Target** for **Package: N/A** to enable it.
      2) Expand **Package: N/A**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: X12V5R1 (ALL)**.
      4) Expand **Protocol: X12V5R1 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: 850 (ALL)**.
   b. Next, enable the document definition representing the envelope:
      1) Click the **Role is not active** icon under **Set Target** for **Package: None** to enable it.
      2) Expand **Package: None**.
      3) Click the **Role is not active** icon under **Set Target** for **Protocol: EDI-X12 (ALL)**.
      4) Expand **Protocol: EDI-X12 (ALL)**.
      5) Click the **Role is not active** icon under **Set Target** for **Document Type: ISA (ALL)**.

## Creating the envelope profile

### About this task

You next create the profile for the envelope that will contain the transformed 850 transaction.

1. Click **Hub Admin > Hub Configuration > EDI > Envelope Profile**.
2. Click **Create**.
3. Type the name of the profile: **EnvProf1**.
4. From the EDI Standard list, select **X12**.
5. The **General** button is selected by default. Type the following values for the general attributes of the envelope:
   - INTCTLLEN: **9**
   - GRPCTLLEN: **9**
   - TRXCTLLEN: **9**
   - MAXDOCS: **1000**
6. Click the **Interchange** button and type the following values for the interchange attributes:

- ISA01: **01**
- ISA02: **ISA0000002**
- ISA03: **02**
- ISA04: **ISA0000004**
- ISA11: \
- ISA12: **00501**
- ISA15: **T**

7. Click **Save**.

# Activating the connections

## About this task

To activate the connections:

1. Click **Account Admin > Connections**.
2. Select **Manager** from the Source list.
3. Select **TP1** from the Target list.
4. Click **Search**.
5. Click **Activate** for the connection that represents the ROD document to EDI transaction:

*Table 50. ROD to EDI connection*

| Source | Target |
|---|---|
| Package: N/A (N/A) Protocol: ROD-TO-EDI_DICT (ALL) Document Type: DTROD-TO-EDI_ROD (ALL) | Package: None (N/A) Protocol: X12V5R1 (ALL) Document Type: 850 |

6. Click **Activate** for the connection that represents the envelope:

*Table 51. Envelope connection*

| Source | Target |
|---|---|
| Package: None (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA (ALL) | Package: N/A (N/A) Protocol: EDI-X12 (ALL) Document Type: ISA(ALL) |

# Configuring attributes

## About this task

To specify attributes for the envelope profile:

1. Click **Account Admin > Profiles > Partner** and click **Search**.
2. Select **TP1** from the list.
3. Click **B2B Capabilities**.
4. Click the **Expand** icon next to **Package: N/A**.
5. Click the **Edit** icon next to **Protocol: X12V5R1**.
6. Specify the following attributes:
   a. In the Envelope Profile row, select **EnvProf1** from the list.
   b. In the Interchange qualifier row, type **01**.
   c. In the Interchange identifier row, type **000000001**.
   d. In the Interchange usage indicator row, type **T**.
7. Click **Save**.

At this point, if the internal partner sent a ROD document to the hub, the document would be transformed to an 850 transaction, which would then be enveloped and sent to the destination of the partner.

# Chapter 21. Additional RosettaNet information

This appendix gives you additional information about RosettaNet support. It includes the following topics:



## Deactivating PIPs

### About this task

After a PIP package has been uploaded into WebSphere Partner Gateway, it cannot be removed. However, you can deactivate the PIP so that it cannot be used.

To deactivate a PIP for all communications with partners, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition**.
2. Expand the document definitions to reveal the Document Type of the PIP you want to disable.
3. In the Status column of the package, click **Enabled**. The Status column now displays **Disabled**, and WebSphere Partner Gateway cannot use the document definition for the PIP.

To deactivate a PIP communication with a specific partner, deactivate the connection to the partner defined for the PIP.

## Providing failure notification

### 0A1 PIP

If a failure occurs during the processing of a PIP message, WebSphere Partner Gateway uses the 0A1 PIP as the mechanism to broadcast the failure to the partner or back-end system that sent the message. For example, say a back-end system initiates a 3A4 PIP. WebSphere Partner Gateway processes the RNSC message and sends a RosettaNet message to a partner. WebSphere Partner Gateway waits for the response to the RosettaNet message until the waiting time reaches the timeout limit. After this occurs, WebSphere Partner Gateway creates a 0A1 PIP and sends it to the partner. The 0A1 PIP identifies the exception condition so that the partner can then compensate for the failure of the 3A4 PIP.

To provide failure notification, upload a 0A1 package and create a PIP connection to the partner using this package.

### Updating contact information

To change the RosettaNet contact information with the 0A1 PIP, you must edit the BCG.Properties file, located in the <*ProductDir*>/router/lib/config directory.

These fields populate the contact information within the 0A1 PIP. Fax is optional (the value can be empty), but the rest are required.

- **bcg.0A1.fromContactName**
- **bcg.0A1.fromEMailAddr**
- **bcg.0A1.fromPhoneNbr**
- **bcg.0A1.fromFaxNbr**

The telephone numbers are limited to 30 bytes in length. The other fields are unlimited in length. When the values are changed, the Document Manager must be restarted.

## Editing RosettaNet attribute values

### About this task

For RosettaNet support, an action type document definition has a specific set of attributes. These attributes provide information used to validate the PIP message, to define the roles and services used in the PIP, and to define the response to the action. The PIP packages provided by WebSphere Partner Gateway automatically define values for these attributes and you typically do not need to change them.

To edit the RosettaNet attributes of an action document definition, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Document Definition.**
2. Click the **Expand** icons to individually expand a node to the appropriate document definition level or select **All** to expand the entire tree.
3. The Actions column for each action contains an **Edit RosettaNet Attribute Values** icon. Click this icon to edit the RosettaNet attributes of the action. The Community Console displays a list of defined attributes under RosettaNet Attributes.
4. Complete the following parameters under RosettaNet Attributes. (These attributes are defined automatically when a PIP is uploaded to the system.)

*Table 52. RosettaNet attributes*

| RosettaNet attribute | Description |
|---|---|
| DTD Name | Identifies the name of the action of the PIP in the DTD provided by RosettaNet |
| From Service | Contains the network component service name of the partner or back-end system that is sending the message |
| To Service | Contains the network component service name of the partner or back-end system that is receiving the message |
| From Role | Contains the role name of the partner or back-end system that is sending the message |
| To Role | Contains the role name of the partner or back-end system that is receiving the message |
| Root Tag | Contains the name of the root element in the XML document associated with the PIP |
| Response From Action Name | Identifies the next Action to perform in the PIP |

**Note:** If the Console displays the `No attributes were found` message, the attributes have not been defined.

5. If the Console displays this message for a lower-level definition, the definition might still work, because it inherits the attributes of the higher-level definition. Adding attributes and their values overrides the inherited attributes and changes the function of the document definition.
6. Click **Save**.

# Creating PIP document definition packages

## About this task

Because RosettaNet adds PIPs from time to time, you might need to create your own PIP packages to support these new PIPs or to support upgrades to PIPs. Except where noted, the procedures in this section describe how to create the PIP document definition package for PIP 5C4 V01.03.00. WebSphere Partner Gateway supplies a PIP document definition package for PIP 5C4 V01.02.00. The procedures, therefore, actually document how to perform an upgrade. However, creating a PIP document definition package is similar and the procedures identify any additional steps.

Before you begin, download the PIP specifications from www.rosettanet.org for the new version, and if you are performing an upgrade, the old version. For example, if you are performing the upgrade described in the procedures, download 5C4_DistributeRegistrationStatus_V01_03_00.zip and 5C4_DistributeRegistrationStatus_V01_02_00.zip. The specification includes the following file types:

- RosettaNet XML Message Guidelines - HTML files such as 5C4_MG_V01_03_00_RegistrationStatusNotification.htm that define the cardinality, vocabulary, structure, and allowable data element values and value types of the PIP.
- RosettaNet XML Message Schema - DTD files such as 5C4_MS_V01_03_RegistrationStatusNotification.dtd that define the order or sequence, element naming, composition, and attributes of the PIP.
- PIP Specification - DOC file such as 5C4_Spec_V01_03_00.doc that provides the business performance controls of the PIP.
- PIP Release Notes - DOC file such as 5C4_V01_03_00_ReleaseNotes.doc that describes the difference between this version and the previous version.

Creating or upgrading a PIP document definition package involves the following procedures:

- Creating the XSD files
- Creating the XML file
- Creating the packages

# Creating the XSD files

## About this task

A PIP document definition package contains XML schema files that define message formats and acceptable values for elements. The following procedure describes how to create these files based on the contents of the PIP specification file.

You create at least one XSD file for each DTD file in the PIP specification file. For the example of upgrading to PIP 5C4 V01.03.00, because the message format changed, the procedure describes how to create the BCG_5C4RegistrationStatusNotification_V01.03.xsd file as an example. For information about the XSD files, see "About validation" on page 362.

To create the XSD files for the PIP document definition package, perform the following steps:

1. Import or load the DTD file into an XML editor such as WebSphere Studio Application Developer. For example, load the 5C4_MS_V01_03_RegistrationStatusNotification.dtd file.
2. Using the XML editor, convert the DTD into an XML schema. The following steps describe how to do this using Application Developer:
   a. In the Navigation pane of the XML perspective, open the project containing the imported DTD file.
   b. Right click the DTD file and select **Generate > XML Schema**.
   c. In the Generate panel, type or select where you want to save the new XSD file. In the File name field, type the name of the new XSD file. In the case of the example, you would type a name such as BCG_5C4RegistrationStatusNotification_V01.03.xsd.
   d. Click **Finish**.
3. Compensate for elements that have multiple cardinality values in the RosettaNet XML guidelines by adding specifications to the new XSD file. The guidelines show the elements in the message using a tree and displaying the cardinality of each element to the left of the element.

   Generally, the elements in the guidelines match the definitions of the elements in the DTD file. However, the guidelines might contain some elements that have the same names but different cardinalities. Because the DTD cannot provide the cardinality in this case, you need to modify the XSD. For example, the 5C4_MG_V01_03_00_RegistrationStatusNotification.htm guidelines file has a definition for ContactInformation on line 15 that has five child elements with the following cardinalities:

   1 contactName

   0..1 EmailAddress

   0..1 facsimileNumber

   0..1 PhysicalLocation

   0..1 telephoneNumber

   The ContactInformation definition on the line 150 has four child elements with the following cardinalities:

   1 contactName

   1 EmailAddress

   0..1 facsimileNumber

   1 telephoneNumber

   In the XSD file, however, each child of ContactInformation has a cardinality that complies with both definitions:

```
<xsd:element name="ContactInformation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="contactName"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="EmailAddress"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="facsimileNumber"/>
      <xsd:element maxOccurs="1" minOccurs="0" ref="PhysicalLocation"/>
```

```
      <xsd:element maxOccurs="1" minOccurs="0" ref="telephoneNumber"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

If you are updating the PIP document definition package based on another version of the package and want to reuse a definition from the other version, perform the following steps for each of these definitions:

a. Delete the definition of the element. For example, delete the ContactInformation element.

b. Open the PIP document definition package of the version being replaced. For example, open the BCG_Package_RNIFV02.00_5C4V01.02.zip file.

c. Find the definition you want to reuse. For example, the ContactInformation_type7 definition in the BCG_ContactInformation_Types.xsd file matches the definition you need for line 15 of the guidelines.

```
<xsd:complexType name="ContactInformation_type7">
  <xsd:sequence>
    <xsd:element name="contactName" type="common_FreeFormText_R"/>
    <xsd:element name="EmailAddress" type="common_EmailAddres_R"
        minOccurs="0"/>
    <xsd:element name="facsimileNumber"
        type="common_CommunicationsNumber_R" minOccurs="0"/>
    <xsd:element name="PhysicalLocation"
        type="PhysicalLocation_type1" minOccurs="0" />
    <xsd:element name="telephoneNumber"
        type="common_CommunicationsNumber_R minOccurs="0" />
  </xsd:sequence>
</xsd:complexType>
```

d. In the new XSD file you are creating for the updated PIP document definition package, create a reference to the XSD file containing the definition you want to reuse. For example, create a reference to BCG_ContactInformation_Types.xsd in the BCG_5C4RegistrationStatusNotification_V01.03.xsd file as follows:

```
<xsd:include schemaLocation="BCG_ContactInformation_Types.xsd"/>
```

e. In the new XSD file, delete the ref attribute of any elements that refer to the element you deleted. Add a type attribute that refers to the definition you are reusing. For example, in the productProviderFieldApplicationEngineer element, delete *ref="Contact Information"* and add the following information:

```
name="ContactInformation"
 type="ContactInformation_type7"
```

If you are creating a PIP document definition package, or are upgrading a PIP document definition package but the definition you need does not exist in the other version, perform the following steps for each instance of the element you found in the guidelines:

a. Delete the definition of the element. For example, delete the ContactInformation element.

b. Create the replacement definition. For example, create the ContactInformation_localType1 definition to match the definition in line 15 of the guidelines.

```
<xsd:complexType name="ContactInformation_localType1">
  <xsd:sequence>
    <xsd:element ref="contactName"/>
    <xsd:element maxOccurs="1" minOccurs="0" ref="EmailAddress"/>
    <xsd:element maxOccurs="1" minOccurs="0"
        ref="facsimileNumber"/>
    <xsd:element maxOccurs="1" minOccurs="0"
        ref="PhysicalLocation"/>
```

```
        <xsd:element maxOccurs="1" minOccurs="0"
            ref="telephoneNumber"/>
      </xsd:sequence>
    </xsd:complexType>
```

c. For any elements that refer to the element you deleted, delete its ref attribute and add a type attribute that refers to the appropriate complex type you defined in the previous step. For example, in the productProviderFieldApplicationEngineer element, delete *ref="Contact Information"* and add the following information:

```
name="ContactInformation"
type="ContactInformation_localType1"
```

Figure 35 shows the productProviderFieldApplicationEngineer element before it is modified.

```
<xsd:element name="productProviderFieldApplicationEngineer">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ContactInformation"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

*Figure 35. Element productProviderFieldApplicationEngineer before modification*

Figure 36 shows the productProviderFieldApplicationEngineer element after it is modified.

```
<xsd:element name="productProviderFieldApplicationEngineer">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ContactInformation"
          type="ContactInformation_localType1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

*Figure 36. Element productProviderFieldApplicationEngineer after modification*

4. Specify the enumeration values for elements that can only have specific values. The guidelines define the enumeration values in the tables in the Guideline Information section.

   For example, in a PIP 5C4 V01.03.00 message, the GlobalRegistrationComplexityLevelCode can have only the following values: Above average, Average, Maximum, Minimum, None, and Some.

   If you are updating the PIP document definition package based on another version of the package and want to reuse a set of enumeration values from the other version, perform the following steps for each set:

   a. Delete the definition for the element. For example, delete the GlobalRegistrationComplexityLevelCode element:

   b. Open the PIP document definition package of the version being replaced. For example, open the BCG_Package_RNIFV02.00_5C4V01.02.zip file.

   c. Find the definition containing the enumeration values you want to reuse. For example, the _GlobalRegistrationComplexityLevelCode definition in the BCG_GlobalRegistrationComplexityLevelCode.xsd file contains the enumeration value definitions defined by the Entity Instance table.

```
<xsd:simpleType name="_GlobalRegistrationComplexityLevelCode">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Above average"/>
    <xsd:enumeration value="Average"/>
```

```
    <xsd:enumeration value="Maximum"/>
    <xsd:enumeration value="Minimum"/>
    <xsd:enumeration value="None"/>
    <xsd:enumeration value="Some"/>
  </xsd:restriction>
</xsd:simpleType>
```

d. In the new XSD file you are creating for the updated PIP document definition package, create a reference to the XSD file containing the definition you want to reuse. For example, create a reference to BCG_GlobalRegistrationComplexityLevelCode.xsd in the BCG_5C4RegistrationStatusNotification_V01.03.xsd file as follows:

```
<xsd:include schemaLocation=
    "BCG_GlobalRegistrationComplexityLevelCode_Types.xsd" />
```

e. In the new XSD file, delete the ref attribute of any elements that refer to the element you deleted. Add a type attribute that refers to the definition you are reusing. For example, in the DesignAssemblyInformation element, delete *ref=*"GlobalRegistrationComplexityLevelCode" and add the following information:

```
name="GlobalRegistrationComplexityLevelCode"
type="_GlobalRegistrationComplexityLevelCode"
```

If you are creating a PIP document definition package or are upgrading a PIP document definition package but the enumeration value definitions you need do not exist in the other version, perform the following steps for any element with enumerated values in the guidelines:

a. Delete the definition of the element. For example, delete the GlobalRegistrationComplexityLevelCode element.

b. Create the replacement definition. For example, create the GlobalRegistrationComplexityLevelCode_localType definition and include the enumeration value definitions as described by the table.

```
<xsd:simpleType
    name="GlobalRegistrationComplexityLevelCode_localType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Above average"/>
    <xsd:enumeration value="Average"/>
    <xsd:enumeration value="Maximum"/>
    <xsd:enumeration value="Minimum"/>
    <xsd:enumeration value="None"/>
    <xsd:enumeration value="Some"/>
  </xsd:restriction>
</xsd:simpleType>
```

c. For any elements that refer to the element you deleted, delete its ref attribute and add a type attribute that refers to the appropriate complex type you defined in the previous step. For example, delete *ref="GlobalRegistrationComplexityLevelCode"* and add the following information:

```
name="GlobalRegistrationComplexityLevelCode"
type="GlobalRegistrationComplexityLevelCode_localType"
```

Figure 37 on page 358 shows the Element DesignAssemblyInformation element before it is modified.

```
<xsd:element name="DesignAssemblyInformation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="0"
          ref="assemblyComments"/>
      <xsd:element maxOccurs="1" minOccurs="0"
          ref="demandCreatorTrackingIdentifier"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0"
          ref="DesignPartInformation"/>
      <xsd:element ref="DesignRegistrationIdentification"/>
      <xsd:element maxOccurs="1" minOccurs="0"
          ref="GeographicRegion"/>
      <xsd:element maxOccurs="1" minOccurs="0"
          ref="GlobalRegistrationComplexityLevelCode"/>
      <xsd:element maxOccurs="1" minOccurs="0"
          ref="GlobalRegistrationInvolvementLevelCode"/>
      <xsd:element maxOccurs="1" minOccurs="0"
          ref="RegistrationStatus"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

*Figure 37. Element DesignAssemblyInformation before modification*

Figure 38 shows the Element DesignAssemblyInformation after it was modified.

```
<xsd:element name="DesignAssemblyInformation">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="1" minOccurs="0"
         ref="assemblyComments"/>
      <xsd:element maxOccurs="1" minOccurs="0"
         ref="demandCreatorTrackingIdentifier"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0"
         ref="DesignPartInformation"/>
      <xsd:element ref="DesignRegistrationIdentification"/>
      <xsd:element maxOccurs="1" minOccurs="0"
         ref="GeographicRegion"/>
      <xsd:element maxOccurs="1" minOccurs="0"
         name="GlobalRegistrationComplexityLevelCode"
           type="GlobalRegistrationComplexityLevelCode_localType"/>

      <xsd:element maxOccurs="1" minOccurs="0"
         ref="GlobalRegistrationInvolvementLevelCode"/>
      <xsd:element maxOccurs="1" minOccurs="0"
         ref="RegistrationStatus"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

*Figure 38. Element DesignAssemblyInformation after modification*

5. Set the data type, minimum length, maximum length, and representation of the data entities. The RosettaNet XML Message Guidelines provide this information in the Fundamental Business Data Entities table.

   If you are updating the PIP document definition package based on another version of the package and want to reuse a data entity definition from the other version, perform the following steps for each set:

   a. Delete the definition for the data entity element. For example, delete the DateStamp element.
   b. Open the PIP document definition package of the version you are replacing. For example, open the BCG_Package_RNIFV02.00_5C4V01.02.zip file.

c. Find the definition you want to reuse. For example, the _common_DateStamp_R definition in the BCG_common.xsd file contains the following definition, which complies with the information given in the guidelines.

```
<xsd:simpleType name="_common_DateStamp_R">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]{8}Z" />
  </xsd:restriction>
</xsd:simpleType>
```

d. In the new XSD file you are creating for the updated PIP document definition package, create a reference to the XSD file containing the definition you want to reuse. For example, create a reference to BCG_common.xsd in the BCG_5C4RegistrationStatusNotification_V01.03.xsd file as follows:

```
<xsd:include schemaLocation="BCG_common.xsd" />
```

e. In the new XSD file, delete the ref attribute of any elements that refer to the element you deleted. Add a type attribute that refers to the definition you are reusing. For example, in the DesignAssemblyInformation element, delete *ref=*"DateStamp" and add the following information:

```
name="DateStamp" type="_common_DateStamp_R"
```

If you are creating a PIP document definition package or are upgrading a PIP document definition package but the data entity definition you need does not exist in the other version, perform the following steps for each data entity element:

a. Delete the definition of the element. For example, delete the DateStamp element.

b. Create the replacement definition. For example, use the data type, minimum length, maximum length, and representation information to create the DateStamp_localType definition.

```
<xsd:simpleType name="DateStamp_localType">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]{8}Z" />
  </xsd:restriction>
</xsd:simpleType>
```

c. For any elements that refer to the element you deleted, delete its ref attribute and add a type attribute that refers to the appropriate complex type you defined in the previous step. For example, delete *ref="DateStamp"* and add the following information:

```
name="DateStamp" type="DateStamp_localType"
```

Figure 39 shows the Element beginDate before it is modified.

```
<xsd:element name="beginDate">
  <xsd:complexType">
    <xsd:sequence>
      <xsd:element ref="DateStamp"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

*Figure 39. Element beginDate before modification*

Figure 40 on page 360 shows the Element beginDate after it is modified.

```
<xsd:element name="beginDate">
  <xsd:complexType">
    <xsd:sequence>
      <xsd:element name="DateStamp" type="DateStamp_localType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

*Figure 40. Element beginDate after modification*

# Creating the XML file

## About this task

After you have created the XSD files for your PIP document definition package, you are ready to create the XML file for the RNIF package and the XML file for the Backend Integration package. For example, these packages are called BCG_Package_RNIFV02.00_5C4V01.03.zip and BCG_Package_RNSC1.0_RNIFV02.00_5C4V01.03.zip. The following procedure describes how to create the XML file for the RNIF package:

1. Extract the XML file from an RNIF PIP document definition package file. If you are upgrading, extract the file from the previous version of the package (for example, BCG_Package_RNIFV02.00_5C4V01.02.zip). If you are creating a new package, extract the file from a PIP document definition package that is similar to the one you are creating. For example, if you are creating a package to support a two-action PIP, copy the XML file from another two-action PIP package.
2. Copy the file and rename it appropriately (for example, BCG_RNIFV02.00_5C4V01.03.xml).
3. In the new file, update the elements that contain information about the PIP. For example, the following table lists the information you need to update in the 5C4 PIP example. Note that the information might appear more than once in the file. Make sure that you update all instances.

*Table 53. 5C4 PIP update information*

| Information to change | Old value | New value |
|---|---|---|
| PIP ID | 5C4 | 5C4 |
| Version of the PIP | V01.02 | V01.03 |
| The name of the request message DTD file without the file extension | 5C4_MS_V01_02_ RegistrationStatusNotification | 5C4_MS_V01_03_ RegistrationStatusNotification |
| The name of the confirmation message DTD file without the file extension (for two-action PIPs only) | N/A | N/A |
| The name of the request message XSD file without the file extension | BCG_5C4RegistrationStatus Notification_V01.02 | BCG_5C4RegistrationStatus Notification_V01.03 |
| The name of the confirmation message XSD file without the file extension (for two-action PIPs only) | N/A | N/A |

*Table 53. 5C4 PIP update information (continued)*

| Information to change | Old value | New value |
|---|---|---|
| Root element name in the XSD file for the request message | Pip5C4RegistrationStatus Notification | Pip5C4RegistrationStatus Notification |
| Root element name in the XSD file for the confirmation message (for two-action PIPs only) | N/A | N/A |

4. Open the PIP Specification document and use it to update the information listed in the following table. If you are doing an update, compare the specifications for the versions because you might not have to update these values.

*Table 54. 5C4 PIP update information from the PIP specification*

| Information to update | Description | Value in the 5C4 package |
|---|---|---|
| Activity name | Specified in Table 3-2 | Distribute Registration Status |
| Initiator role name | Specified in Table 3-1 | Product Provider |
| Responder role name | Specified in Table 3-1 | Demand Creator |
| Request action name | Specified in Table 4-2 | Registration Status Notification |
| Confirmation action name | Specified in Table 4-2 (for two-action PIPs only) | N/A |

5. Update the package attribute values. If you are doing an update, compare the specifications for the versions because you might not have to update these values.

   **Note:** If you are creating the Backend Integration package, skip this step and go to step 6 on page 362.

*Table 55. 5C4 PIP attribute updates*

| Information to update | Description | Value in the 5C4 package | Element path in the XML file |
|---|---|---|---|
| NonRepudiation Required | Specified in Table 3-3 | N | ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is NonRepudiationRequired) ns1:AttributeValue AttributePickListItem ATTRVALUEKEY |
| NonRepudiationOf Receipt | Specified in Table 3-3 | N | ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is NonRepudiationOfReceipt) ns1:AttributeValue AttributePickListItem ATTRVALUEKEY |
| DigitalSignature Required | Specified in Table 5-1 | Y | ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is DigitalSignatureRequired) ns1:AttributeValue AttributePickListItem ATTRVALUEKEY |

*Table 55. 5C4 PIP attribute updates (continued)*

| Information to update | Description | Value in the 5C4 package | Element path in the XML file |
|---|---|---|---|
| TimeToAcknowledge | Specified in Table 3-3 | 2 (120 min) | ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is TimeToAcknowledge) ns1:AttributeValue ATTRVALUE |
| TimeToPerform | Specified in Table 3-3 | 2 (120 min) | ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is TimeToPerform) ns1:AttributeValue ATTRVALUE |
| RetryCount | Specified in Table 3-3 | 3 | ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is RetryCount) ns1:AttributeValue ATTRVALUE |

6. Update the ns1:Package/ns1:Protocol/GuidelineMap elements to remove unused XSD files and to add any XSD files you created or referenced.

To create the Backend Integration package, repeat step 1 on page 360 through 6 except for the following differences:

- In step 1 on page 360, extract the XML file from the Backend Integration package (for example, BCG_Package_RNSC1.0_RNIFV02.00_5C4V01.02.zip).
- Do not do step 5 on page 361.

After you have created the XML and the XSD files, you are ready to create the PIP documentation flow packages.

## Creating the package

### About this task

To create the RNIF package, perform the following steps:

1. Create a GuidelineMaps directory and copy the package's XSD files into this directory.
2. Create a Packages directory and copy the RNIF XML file into this directory.
3. Go to the parent directory and create a PIP document definition package (ZIP file) that contains the GuidelineMaps and Packages directory. You must preserve the directory structure in the ZIP file.

To create the Backend Integration package, perform steps 1 through 3 but use the Backend Integration XML file instead of the RNIF file.

After you have created the PIP package, you can upload it using the procedure in "RNIF and PIP document type packages" on page 106.

# About validation

WebSphere Partner Gateway validates the service content of a RosettaNet message using validation maps. These validation maps define the structure of a valid message and define the cardinality, format, and valid values (enumeration) of the elements within the message. Within each PIP document definition package, WebSphere Partner Gateway supplies the validation maps as XSD files in the GuidelineMaps directory.

Because RosettaNet specifies the format of a PIP message, typically you will not need to customize the validation maps. However, if you do, see “Creating PIP document definition packages” on page 353 for information about the steps needed to upgrade the XSD files used to validate the messages and how to create a custom PIP document definition package.

## Cardinality

Cardinality determines the number of times a particular element can or must appear in a message. In the validation maps, the minOccurs and maxOccurs attributes determine the cardinality of the attribute as shown in the following example taken from BCG_5C4RegistrationStatusNotification_V01.02.xsd:

```
<xsd:element name="GeographicRegion" type="GeographicRegionType"
    minOccurs="0"/>
```

If WebSphere Partner Gateway does not need to check the cardinality of an element, the values of the element's minOccurs and maxOccurs attributes in the validation map are "0" and "unbounded", as shown in the following example:

```
<xsd:element name="DesignRegistrationIdentification"
    type="DesignRegistrationIdentificationType2"
    minOccurs="0" maxOccurs="unbounded"/>
```

## Format

Format determines the arrangement or layout of data for the type of an element. In the validation maps, the type has one or more restrictions as shown in the following examples:

### Example 1

```
<xsd:simpleType name="_common_LineNumber_R">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1" />
    <xsd:maxLength value="6" />
  </xsd:restriction>
</xsd:simpleType>
```

All _common_LineNumber_R type elements in a message must be Strings and must be 1 to 6 characters in length.

### Example 2

```
<xsd:simpleType name="_GlobalLocationIdentifier">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[0-9]{9}.{1,4}" />
  </xsd:restriction>
</xsd:simpleType>
```

All _GlobalLocationIdentifier type elements in a message must be Strings and must have nine characters of numeric data followed by one to four characters of alphanumeric data. The minimum length is therefore 10 characters and the maximum is 13.

### Example 3

```
<xsd:element name="DayOfMonth">
  <xsd:simpleType>
    <xsd:restriction base="xsd:positiveInteger">
      <xsd:totalDigits value="2" />
      <xsd:minInclusive value="1" />
```

```
      <xsd:maxInclusive value="31" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

All _DayofMonth type elements in a message must be PositiveInteger, must have one or two characters, and have a value of 1 to 31 inclusive.

### Enumeration

Enumeration determines the valid values for an element. In the validation maps, the type of the element has one or more enumeration restrictions as shown in the following example:

```
<xsd:simpleType name="_local_GlobalDesignRegistrationNotificationCode">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Initial" />
    <xsd:enumeration value="Update" />
  </xsd:restriction>
</xsd:simpleType>
```

All _local_GlobalDesignRegistrationNotificationCode type elements in a message must have only "Initial" or "Update" for their values.

## PIP document definition packages

The following sections show the PIP document definition packages provided by WebSphere Partner Gateway for each PIP. Within each package is an XML file contained in a Packages directory and several XSD files contained in a GuidelineMaps directory, which are common to all PIP document definition packages for the PIP.

### 0A1 Notification of Failure V1.0

The following section describes the contents for the 0A1 Notification of Failure V1.0 PIP.

#### Package file contents

The following table shows the ZIP files and corresponding XML files for the 0A1 Notification of Failure V1.0 PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 56. 0A1 Notification of Failure V1.0 PIP ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_0A11.0.zip | BCG_RNIF1.1_0A11.0.xml |
| BCG_Package_RNSC1.0_RNIF1.1_0A11.0.zip | BCG_RNSC1.0_RNIF1.1_0A11.0.xml |

#### Guideline map contents

This section lists the guideline maps contents for 0A1 Notification of Failure V1.0:

- 0A1FailureNotification_1.0.xml
- BCG_0A1FailureNotification_1.0.xsd
- BCG_common.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd

- BCG_string_len_0.xsd
- BCG_xml.xsd

## 0A1 Notification of Failure V02.00

The following section describes the contents for the 0A1 Notification of Failure V02.00 PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 0A1 Notification of Failure V02.00 PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 57. 0A1 Notification of Failure V02.00 PIP ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIFV02.00_0A1V02.00.zip | BCG_RNIFV02.00_0A1V02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_0A1V02.00.zip | BCG_RNSC1.0_RNIFV02.00_0A1V02.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 0A1 Notification of Failure V02.00:

- 0A1FailureNotification_V02.00.xml
- BCG_0A1FailureNotification_V02.00.xsd
- BCG_common.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 2A1 Distribute New Product Information

The following section describes the contents for the 2A1 Distribute New Product Information PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 2A1 Distribute New Product Information PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 58. 2A1 Distribute New Product Information ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_2A1V02.00.zip | BCG_RNIF1.1_2A1V02.00.xml |
| BCG_Package_RNIFV02.00_2A1V02.00.zip | BCG_RNIFV02.00_2A1V02.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_2A1V02.00.zip | BCG_RNSC1.0_RNIF1.1_2A1V02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_2A1V02.00.zip | BCG_RNSC1.0_RNIFV02.00_2A1V02.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 2A1 Distribute New Product Information:

- BCG_2A1ProductCatalogInformationNotification_V02.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V422.xsd
- BCG_BusinessDescription_Types_V43.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_common_V43.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ContactInformation_Types_V422.xsd
- BCG_ContactInformation_Types_V43.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalIntervalCode.xsd
- BCG_GlobalLeadTimeClassificationCode_V43.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPackageTypeCode_V422.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd
- BCG_GlobalPartnerRoleClassificationCode_V422.xsd
- BCG_GlobalPriceTypeCode_V43.xsd
- BCG_GlobalProductAssociationCode_V43.xsd
- BCG_GlobalProductLifeCycleStatusCode.xsd
- BCG_GlobalProductProcurementTypeCode_V43.xsd
- BCG_GlobalProductTypeCode_V43.xsd
- BCG_GlobalProductUnitofMeasureCode_V43.xsd
- BCG_GlobalProprietaryProductIdentificationTypeCode_V43.xsd
- BCG_GlobalStandardClassificationSchemeCode_V43.xsd
- BCG_GlobalWarrantyTypeCode_V43.xsd
- BCG_InvoiceChargeTypeCode_V422.xsd
- BCG_NationalExportControlClassificationCode.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_PhysicalAddress_Types_V422.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 2A12 Distribute Product Master

The following section describes the contents for the 2A12 Distribute Product Master PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 2A12 Distribute Product Master PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 59. 2A12 Distribute Product Master ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_2A12V01.03.zip | BCG_RNIF1.1_2A12V01.03.xml |
| BCG_Package_RNIFV02.00_2A12V01.03.zip | BCG_RNIFV02.00_2A12V01.03.xml |
| BCG_Package_RNSC1.0_RNIF1.1_2A12V01.03.zip | BCG_RNSC1.0_RNIF1.1_2A12V01.03.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_2A12V01.03.zip | BCG_RNSC1.0_RNIFV02.00_2A12V01.03.xml |

### Guideline map contents

This section lists the guideline maps contents for 2A12 Distribute Product Master:

- BCG_2A12ProductMasterNotification_V01.03.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAssemblyLevelCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalIntervalCode.xsd
- BCG_GlobalLeadTimeClassificationCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalProductLifeCycleStatusCode.xsd
- BCG_GlobalProductProcurementTypeCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A1 Request Quote

The following section describes the contents for the 3A1 Request Quote PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A1 Request Quote PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 60. 3A1 Request Quote PIP ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A1V02.00.zip | BCG_RNIF1.1_3A1V02.00.xml |
| BCG_Package_RNIFV02.00_3A1V02.00.zip | BCG_RNIFV02.00_3A1V02.00.xml |

*Table 60. 3A1 Request Quote PIP ZIP and XML files (continued)*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNSC1.0_RNIF1.1_3A1V02.00.zip | BCG_RNSC1.0_RNIF1.1_3A1V02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A1V02.00.zip | BCG_RNSC1.0_RNIFV02.00_3A1V02.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3A1 Request Quote:

- BCG_3A1QuoteConfirmation_V02.00.xsd
- BCG_3A1QuoteRequest_V02.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductTermsCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalQuoteLineItemStatusCode.xsd
- BCG_GlobalQuoteTypeCode.xsd
- BCG_GlobalStockIndicatorCode.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A2 Request Price and Availability

The following section describes the contents for the 3A2 Request Price and Availability PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A2 Request Price and Availability PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 61. 3A2 Request Price and Availability ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A2R02.01.zip | BCG_RNIF1.1_3A2R02.01.xml |
| BCG_Package_RNIFV02.00_3A2R02.01.zip | BCG_RNIFV02.00_3A2R02.01.xml |

*Table 61. 3A2 Request Price and Availability ZIP and XML files (continued)*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNSC1.0_RNIF1.1_3A2R02.01.zip | BCG_RNSC1.0_RNIF1.1_3A2R02.01.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A2R02.01.zip | BCG_RNSC1.0_RNIFV02.00_3A2R02.01.xml |

### Guideline map contents

This section lists the guideline maps contents for 3A2 Request Price and Availability:

- BCG_3A2PriceAndAvailabilityRequest_R02.01.xsd
- BCG_3A2PriceAndAvailabilityResponse_R02.01.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalCustomerAuthorizationCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPricingTypeCode.xsd
- BCG_GlobalProductAvailabilityCode.xsd
- BCG_GlobalProductStatusCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A4 Request Purchase Order V02.00

The following section describes the contents for the 3A4 Request Purchase OrderV02.00 PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A4 Request Purchase Order PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 62. 3A4 Request Purchase Order ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A4V02.00.zip | BCG_RNIF1.1_3A4V02.00.xml |
| BCG_Package_RNIFV02.00_3A4V02.00.zip | BCG_RNIFV02.00_3A4V02.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A4V02.00.zip | BCG_RNSC1.0_RNIF1.1_3A4V02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A4V02.00.zip | BCG_RNSC1.0_RNIFV02.00_3A4V02.00.xml |

## Guideline map contents

This section lists the guideline maps contents for 3A4 Request Purchase Order:

- BCG_3A4PurchaseOrderConfirmation_V02.00.xsd
- BCG_3A4PurchaseOrderRequest_V02.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V422.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAccountClassificationCode.xsd
- BCG_GlobalConfirmationTypeCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCreditCardClassificationCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalDocumentReferenceTypeCode_V422.xsd
- BCG_GlobalFinanceTermsCode.xsd
- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode_V422.xsd
- BCG_GlobalPaymentConditionCode.xsd
- BCG_GlobalPriceUnitOfMeasureCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalPurchaseOrderAcknowledgmentReasonCode.xsd
- BCG_GlobalPurchaseOrderFillPriorityCode.xsd
- BCG_GlobalPurchaseOrderStatusCode.xsd
- BCG_GlobalPurchaseOrderTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShipmentTermsCode_V422.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialFulfillmentRequestCode.xsd
- BCG_GlobalSpecialHandlingCode_V422.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_GlobalTaxExemptionCode_V422.xsd
- BCG_InvoiceChargeTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_PhysicalAddress_Types_V422.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

# 3A4 Request Purchase Order V02.02

The following section describes the contents for the 3A4 Request Purchase OrderV02.02 PIP.

## Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A4 Request Purchase Order PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 63. 3A4 Request Purchase Order ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A4V02.02.zip | BCG_RNIF1.1_3A4V02.02.xml |
| BCG_Package_RNIFV02.00_3A4V02.02.zip | BCG_RNIFV02.00_3A4V02.02.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A4V02.02.zip | BCG_RNSC1.0_RNIF1.1_3A4V02.02.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A4V02.02.zip | BCG_RNSC1.0_RNIFV02.00_3A4V02.02.xml |

## Guideline map contents

This section lists the guideline maps contents for 3A4 Request Purchase Order:

- BCG_3A4PurchaseOrderConfirmation_V02.02.xsd
- BCG_3A4PurchaseOrderRequest_V02.02.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAccountClassificationCode.xsd
- BCG_GlobalConfirmationTypeCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCreditCardClassificationCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFinanceTermsCode.xsd
- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPaymentConditionCode.xsd
- BCG_GlobalPriceUnitOfMeasureCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalPurchaseOrderAcknowledgmentReasonCode.xsd
- BCG_GlobalPurchaseOrderFillPriorityCode.xsd
- BCG_GlobalPurchaseOrderStatusCode.xsd
- BCG_GlobalPurchaseOrderTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd

- BCG_GlobalSpecialFulfillmentRequestCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_InvoiceChargeTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A5 Query Order Status

The following section describes the contents for the 3A5 Query Order Status PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A5 Query Order Status PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 64. 3A5 Query Order Status ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A5R02.00.zip | BCG_RNIF1.1_3A5R02.00.xml |
| BCG_Package_RNIFV02.00_3A5R02.00.zip | BCG_RNIFV02.00_3A5R02.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A5R02.00.zip | BCG_RNSC1.0_RNIF1.1_3A5R02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A5R02.00.zip | BCG_RNSC1.0_RNIFV02.00_3A5R02.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3A5 Query Order Status:

- BCG_3A5PurchaseOrderStatusQuery_R02.00.xsd
- BCG_3A5PurchaseOrderStatusResponse_R02.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAccountClassificationCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCreditCardClassificationCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalCustomerTypeCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFinanceTermsCode.xsd
- BCG_GlobalFreeOnBoardCode.xsd
- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalLineItemStatusCode.xsd
- BCG_GlobalOrderQuantityTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd

- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalPurchaseOrderAcknowledgmentReasonCode.xsd
- BCG_GlobalPurchaseOrderFillPriority
- BCG_GlobalPurchaseOrderStatusCode.xsd
- BCG_GlobalPurchaseOrderTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialFulfillmentRequestCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_GlobalTransportEventCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A6 Distribute Order Status

The following section describes the contents for the 3A6 Distribute Order Status PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A6 Distribute Order Status PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 65. 3A6 Distribute Order Status ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A6V02.02.zip | BCG_RNIF1.1_3A6V02.02.xml |
| BCG_Package_RNIFV02.00_3A6V02.02.zip | BCG_RNIFV02.00_3A6V02.02.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A6V02.02.zip | BCG_RNSC1.0_RNIF1.1_3A6V02.02.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A6V02.02.zip | BCG_RNSC1.0_RNIFV02.00_3A6V02.02.xml |

### Guideline map contents

This section lists the guideline maps contents for 3A6 Distribute Order Status:

- BCG_3A6PurchaseOrderStatusNotification_V02.02.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAccountClassificationCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCreditCardClassificationCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd

- BCG_GlobalFinanceTermsCode.xsd
- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalLineItemStatusCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalNotificationReasonCode.xsd
- BCG_GlobalOrderQuantityTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPaymentConditionCode.xsd
- BCG_GlobalPriceUnitOfMeasureCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalPurchaseOrderAcknowledgmentReasonCode.xsd
- BCG_GlobalPurchaseOrderFillPriorityCode.xsd
- BCG_GlobalPurchaseOrderStatusCode.xsd
- BCG_GlobalPurchaseOrderTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialFulfillmentRequestCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_GlobalTrackingReferenceTypeCode.xsd
- BCG_InvoiceChargeTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A7 Notify of Purchase Order Update

The following section describes the contents for the 3A7 Notify of Purchase Order Update PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A7 Notify of Purchase Order Update PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 66. 3A7 Notify of Purchase Order Update ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A7V02.02.zip | BCG_RNIF1.1_3A7V02.02.xml |
| BCG_Package_RNIFV02.00_3A7V02.02.zip | BCG_RNIFV02.00_3A7V02.02.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A7V02.02.zip | BCG_RNSC1.0_RNIF1.1_3A7V02.02.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A7V02.02.zip | BCG_RNSC1.0_RNIFV02.00_3A7V02.02.xml |

### Guideline map contents

This section lists the guideline maps contents for 3A7 Notify of Purchase Order Update:

- BCG_3A7PurchaseOrderUpdateNotification_V02.02.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAccountClassificationCode.xsd
- BCG_GlobalActionCode.xsd
- BCG_GlobalConfirmationTypeCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCreditCardClassificationCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFinanceTermsCode.xsd
- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPaymentConditionCode.xsd
- BCG_GlobalPriceUnitOfMeasureCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalPurchaseOrderAcknowledgmentReasonCode.xsd
- BCG_GlobalPurchaseOrderFillPriorityCode.xsd
- BCG_GlobalPurchaseOrderStatusCode.xsd
- BCG_GlobalPurchaseOrderTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialFulfillmentRequestCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_InvoiceChargeTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A8 Request Purchase Order Change V01.02

The following section describes the contents for the 3A8 Request Purchase Order Change V01.02 PIP.

## Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A8 Request Purchase Order Change PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 67. 3A8 Request Purchase Order Change ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A8V01.02.zip | BCG_RNIF1.1_3A8V01.02.xml |
| BCG_Package_RNIFV02.00_3A8V01.02.zip | BCG_RNIFV02.00_3A8V01.02.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A8V01.02.zip | BCG_RNSC1.0_RNIF1.1_3A8V01.02.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A8V01.02.zip | BCG_RNSC1.0_RNIFV02.00_3A8V01.02.xml |

## Guideline map contents

This section lists the guideline maps contents for 3A8 Request Purchase Order Change:

- BCG_3A8PurchaseOrderChangeConfirmation_V01.02.xsd
- BCG_3A8PurchaseOrderChangeRequest_V01.02.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAccountClassificationCode.xsd
- BCG_GlobalActionCode.xsd
- BCG_GlobalConfirmationTypeCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCreditCardClassificationCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFinanceTermsCode.xsd
- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPaymentConditionCode.xsd
- BCG_GlobalPriceUnitOfMeasureCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalPurchaseOrderAcknowledgmentReasonCode.xsd
- BCG_GlobalPurchaseOrderFillPriorityCode.xsd
- BCG_GlobalPurchaseOrderStatusCode.xsd
- BCG_GlobalPurchaseOrderTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialFulfillmentRequestCode.xsd

- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_InvoiceChargeTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A8 Request Purchase Order Change V01.03

The following section describes the contents for the 3A8 Request Purchase Order Change V01.03 PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A8 Request Purchase Order Change PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 68. 3A8 Request Purchase Order Change ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A8V01.03.zip | BCG_RNIF1.1_3A8V01.03.xml |
| BCG_Package_RNIFV02.00_3A8V01.03.zip | BCG_RNIFV02.00_3A8V01.03.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A8V01.03.zip | BCG_RNSC1.0_RNIF1.1_3A8V01.03.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A8V01.03.zip | BCG_RNSC1.0_RNIFV02.00_3A8V01.03.xml |

### Guideline map contents

This section lists the guideline maps contents for 3A8 Request Purchase Order Change:

- BCG_3A8PurchaseOrderChangeConfirmation_V01.03.xsd
- BCG_3A8PurchaseOrderChangeRequest_V01.03.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V43.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_common_V43.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAccountClassificationCode.xsd
- BCG_GlobalActionCode.xsd
- BCG_GlobalConfirmationTypeCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCreditCardClassificationCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFinanceTermsCode.xsd
- BCG_GlobalFreeOnBoardCode_V422.xsd

- BCG_GlobalGovernmentPriorityRatingCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd
- BCG_GlobalPartnerRoleClassificationCode_V43.xsd
- BCG_GlobalPaymentConditionCode.xsd
- BCG_GlobalProductSubstitutionReasonCode.xsd
- BCG_GlobalProductUnitOfMeasureCode_V43.xsd
- BCG_GlobalPurchaseOrderAcknowledgmentReasonCode.xsd
- BCG_GlobalPurchaseOrderFillPriorityCode.xsd
- BCG_GlobalPurchaseOrderStatusCode.xsd
- BCG_GlobalPurchaseOrderTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialFulfillmentRequestCode.xsd
- BCG_GlobalSpecialHandlingCode_V43.xsd
- BCG_GlobalTaxExemptionCode.xsd
- BCG_GlobalTransportEventCode.xsd
- BCG_InvoiceChargeTypeCode_V422.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3A9 Request Purchase Order Cancellation

The following section describes the contents for the 3A9 Request Purchase Order Cancellation PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3A9 Request Purchase Order Cancellation PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 69. 3A9 Request Purchase Order Cancellation ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3A9V01.01.zip | BCG_RNIF1.1_3A9V01.01.xml |
| BCG_Package_RNIFV02.00_3A9V01.01.zip | BCG_RNIFV02.00_3A9V01.01.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3A9V01.01.zip | BCG_RNSC1.0_RNIF1.1_3A9V01.01.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3A9V01.01.zip | BCG_RNSC1.0_RNIFV02.00_3A9V01.01.xml |

### Guideline map contents

This section lists the guideline maps contents for 3A9 Request Purchase Order Cancellation:

- BCG_3A9PurchaseOrderCancellationConfirmation_V01.01.xsd
- BCG_3A9PurchaseOrderCancellationRequest_V01.01.xsd

- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPurchaseOrderCancellationCode.xsd
- BCG_GlobalPurchaseOrderCancellationResponseCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3B2 Notify of Advance Shipment

The following section describes the contents for the 3B2 Notify of Advance Shipment PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3B2 Notify of Advance Shipment PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 70. 3B2 Notify of Advance Shipment ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3B2V01.01.zip | BCG_RNIF1.1_3B2V01.01.xml |
| BCG_Package_RNIFV02.00_3B2V01.01.zip | BCG_RNIFV02.00_3B2V01.01.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3B2V01.01.zip | BCG_RNSC1.0_RNIF1.1_3B2V01.01.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3B2V01.01.zip | BCG_RNSC1.0_RNIFV02.00_3B2V01.01.xml |

### Guideline map contents

This section lists the guideline maps contents for 3B2 Notify of Advance Shipment:

- BCG_3B2AdvanceShipmentNotification_V01.01.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalIncotermsCode.xsd
- BCG_GlobalLotQuantityClassificationCode.xsd
- BCG_GlobalPackageTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPhysicalUnitOfMeasureCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalShipDateCode.xsd

- BCG_GlobalShipmentChangeDispositionCode.xsd
- BCG_GlobalShipmentModeCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalTrackingReferenceTypeCode.xsd
- BCG_NationalExportControlClassificationCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3B3 Distribute Shipment Status

The following section describes the contents for the 3B3 Distribute Shipment Status PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3B3 Distribute Shipment Status PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 71. 3B3 Distribute Shipment Status ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3B3R01.00.zip | BCG_RNIF1.1_3B3R01.00.xml |
| BCG_Package_RNIFV02.00_3B3R01.00.zip | BCG_RNIFV02.00_3B3R01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3B3R01.00.zip | BCG_RNSC1.0_RNIF1.1_3B3R01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3B3R01.00.zip | BCG_RNSC1.0_RNIFV02.00_3B3R01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3B3 Distribute Shipment Status:

- 3B3 Distribute Shipment Status_R01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V422.xsd
- BCG_BusinessDescription_Types_V43.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common_V422.xsd
- BCG_common_V43.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ContactInformation_Types_V422.xsd
- BCG_ContactInformation_Types_V43.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode_V43.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd
- BCG_GlobalShipmentDispositionCode.xsd
- BCG_GlobalShipmentModeCode.xsd
- BCG_GlobalShipmentStatusCode_V43.xsd
- BCG_GlobalShipmentStatusReportingLevelCode_V43.xsd

- BCG_GlobalTrackingReferenceTypeCode.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_PhysicalAddress_Types_V422.xsd
- BCG_PhysicalAddress_Types_V423.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3B11 Notify of Shipping Order

The following section describes the contents for the 3B11 Notify of Shipping Order PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3B11 Notify of Shipping Order PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 72. 3B11 Notify of Shipping Order ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3B11R01.00A.zip | BCG_RNIF1.1_3B11R01.00A.xml |
| BCG_Package_RNIFV02.00_3B11R01.00A.zip | BCG_RNIFV02.00_3B11R01.00A.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3B11R01.00A.zip | BCG_RNSC1.0_RNIF1.1_3B11R01.00A.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3B11R01.00A.zip | BCG_RNSC1.0_RNIFV02.00_3B11R01.00A.xml |

### Guideline map contents

This section lists the guideline maps contents for 3B11 Notify of Shipping Order:

- 3B11 ShippingOrderNotification_R01.00A.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V422.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ContactInformation_Types_V422.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode_V422.xsd
- BCG_GlobalFreightPaymentTermsCode_V422.xsd
- BCG_GlobalIncotermsCode.xsd
- BCG_GlobalOrderAdminCode_V422.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalShipDateCode.xsd
- BCG_GlobalShipmentModeCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd

- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalWarrantyTypeCode_V43.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PartnerDescription_Types_V422.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3B12 Request Shipping Order

The following section describes the contents for the 3B12 Request Shipping Order PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3B12 Request Shipping Order PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 73. 3B12 Request Shipping Order ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3B12V01.01.zip | BCG_RNIF1.1_3B12V01.01.xml |
| BCG_Package_RNIFV02.00_3B12V01.01.zip | BCG_RNIFV02.00_3B12V01.01.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3B12V01.01.zip | BCG_RNSC1.0_RNIF1.1_3B12V01.01.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3B12V01.01.zip | BCG_RNSC1.0_RNIFV02.00_3B12V01.01.xml |

### Guideline map contents

This section lists the guideline maps contents for 3B12 Request Shipping Order:

- BCG_3B12ShippingOrderConfirmation_V01.01.xsd
- BCG_3B12ShippingOrderRequest_V01.01.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ContactInformation_Types_V422.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalIncotermsCode.xsd
- BCG_GlobalPackageTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPhysicalUnitOfMeasureCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalShipDateCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd

- BCG_PartnerDescription_Types.xsd
- BCG_PartnerDescription_Types_V422.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3B13 Notify of Shipping Order Confirmation

The following section describes the contents for the 3B13 Notify of Shipping Order Confirmation PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3B13 Notify of Shipping Order Confirmation PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 74. 3B13 Notify of Shipping Order Confirmation ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3B13V01.01.zip | BCG_RNIF1.1_3B13V01.01.xml |
| BCG_Package_RNIFV02.00_3B13V01.01.zip | BCG_RNIFV02.00_3B13V01.01.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3B13V01.01.zip | BCG_RNSC1.0_RNIF1.1_3B13V01.01.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3B13V01.01.zip | BCG_RNSC1.0_RNIFV02.00_3B13V01.01.xml |

### Guideline map contents

This section lists the guideline maps contents for 3B13 Notify of Shipping Order Confirmation:

- BCG_3B13ShippingOrderConfirmationNotification_V01.01.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPhysicalUnitOfMeasureCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalShipDateCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalTrackingReferenceTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd

- BCG_xml.xsd

## 3B14 Request Shipping Order Cancellation

The following section describes the contents for the 3B14 Request Shipping Order Cancellation PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3B14 Request Shipping Order Cancellation PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 75. 3B14 Request Shipping Order Cancellation ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3B14V01.00.zip | BCG_RNIF1.1_3B14V01.00.xml |
| BCG_Package_RNIFV02.00_3B14V01.00.zip | BCG_RNIFV02.00_3B14V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3B14V01.00.zip | BCG_RNSC1.0_RNIF1.1_3B14V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3B14V01.00.zip | BCG_RNSC1.0_RNIFV02.00_3B14V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3B14 Request Shipping Order Cancellation:

- 3B14_ShippingOrderCancellationConfirmation_V01.00.xsd
- 3B14_ShippingOrderCancellationRequest_V01.00.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalOrderAdminCode_V22.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode_V43.xsd
- BCG_GlobalShippingOrderCancellationStatusReasonCode_V43.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3B18 Notify of Shipping Documentation

The following section describes the contents for the 3B18 Notify of Shipping Documentation PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3B18 Notify of Shipping Documentation PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 76. 3B18 Notify of Shipping Documentation ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3B18V01.00.zip | BCG_RNIF1.1_3B18V01.00.xml |
| BCG_Package_RNIFV02.00_3B18V01.00.zip | BCG_RNIFV02.00_3B18V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3B18V01.00.zip | BCG_RNSC1.0_RNIF1.1_3B18V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3B18V01.00.zip | BCG_RNSC1.0_RNIFV02.00_3B18V01.00.xml |

## Guideline map contents

This section lists the guideline maps contents for 3B18 Notify of Shipping Documentation:

- BCG_3B18ShippingDocumentationNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V422.xsd
- BCG_common_V422.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFreeOnBoardCode_V422.xsd
- BCG_GlobalFreightPaymentTermsCode_V422.xsd
- BCG_GlobalIncotermsCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalOrderAdminCode_V422.xsd
- BCG_GlobalPackageTypeCode_V422.xsd
- BCG_GlobalPartnerClassificationCode_V422.xsd
- BCG_GlobalPartnerRoleClassificationCode_V422.xsd
- BCG_GlobalPaymentTermsCode.xsd
- BCG_GlobalPhysicalUnitOfMeasureCode_V422.xsd
- BCG_GlobalPortIdentifierAuthorityCode_V422.xsd
- BCG_GlobalPortTypeCode_V422.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalShipDateCode.xsd
- BCG_GlobalShipmentModeCode.xsd
- BCG_GlobalShippingDocumentCode_V422.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialHandlingCode_V422.xsd
- BCG_GlobalTrackingReferenceTypeCode.xsd
- BCG_InvoiceChargeTypeCode_V422.xsd
- BCG_NationalExportControlClassificationCode.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3C1 Return Product

The following section describes the contents for the 3C1 Return Product PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3C1 Return Product PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 77. 3C1 Return Product ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3C1V01.00.zip | BCG_RNIF1.1_3C1V01.00.xml |
| BCG_Package_RNIFV02.00_3C1V01.00.zip | BCG_RNIFV02.00_3C1V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3C1V01.00.zip | BCG_RNSC1.0_RNIF1.1_3C1V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3C1V01.00.zip | BCG_RNSC1.0_RNIFV02.00_3C1V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3C1 Return Product:

- BCG_3C1ReturnProductConfirmation_V01.00.xsd
- BCG_3C1ReturnProductRequest_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V43.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_common_V43.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFailureTypeCode_V43.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd
- BCG_GlobalPartnerRoleClassificationCode_V43.xsd
- BCG_GlobalProductUnitOfMeasureCode_V43.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3C3 Notify of Invoice

The following section describes the contents for the 3C3 Notify of Invoice PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3C3 Notify of Invoice PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 78. 3C3 Notify of Invoice ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3C3V01.01.zip | BCG_RNIF1.1_3C3V01.01.xml |
| BCG_Package_RNIFV02.00_3C3V01.01.zip | BCG_RNIFV02.00_3C3V01.01.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3C3V01.01.zip | BCG_RNSC1.0_RNIF1.1_3C3V01.01.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3C3V01.01.zip | BCG_RNSC1.0_RNIFV02.00_3C3V01.01.xml |

### Guideline map contents

This section lists the guideline maps contents for 3C3 Notify of Invoice:

- BCG_3C3InvoiceNotification_V01.01.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentTypeCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPaymentTermsCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalSaleTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd
- BCG_InvoiceChargeTypeCode.xsd
- BCG_NationalExportControlClassificationCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3C4 Notify of Invoice Reject

The following section describes the contents for the 3C4 Notify of Invoice Reject PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3C4 Notify of Invoice Reject PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 79. 3C4 Notify of Invoice Reject ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3C4V01.00.zip | BCG_RNIF1.1_3C4V01.00.xml |
| BCG_Package_RNIFV02.00_3C4V01.00.zip | BCG_RNIFV02.00_3C4V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3C4V01.00.zip | BCG_RNSC1.0_RNIF1.1_3C4V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3C4V01.00.zip | BCG_RNSC1.0_RNIFV02.00_3C4V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3C4 Notify of Invoice Reject:

- BCG_3C4InvoiceRejectNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalInvoiceRejectionCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3C6 Notify of Remittance Advice

The following section describes the contents for the 3C6 Notify of Remittance Advice PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3C6 Notify of Remittance Advice PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 80. 3C6 Notify of Remittance Advice ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3C6V01.00.zip | BCG_RNIF1.1_3C6V01.00.xml |
| BCG_Package_RNIFV02.00_3C6V01.00.zip | BCG_RNIFV02.00_3C6V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3C6V01.00.zip | BCG_RNSC1.0_RNIF1.1_3C6V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3C6V01.00.zip | BCG_RNSC1.0_RNIFV02.00_3C6V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3C6 Notify of Remittance Advice:

- BCG_3C6RemittanceAdviceNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd

- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentTypeCode.xsd
- BCG_GlobalFinancialAdjustmentReasonCode.xsd
- BCG_GlobalInvoiceRejectionCode.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPaymentMethodCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3C7 Notify of Self-Billing Invoice

The following section describes the contents for the 3C7 Notify of Self-Billing Invoice PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3C7 Notify of Self-Billing Invoice PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 81. 3C7 Notify of Self-Billing Invoice ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3C7V01.00.zip | BCG_RNIF1.1_3C7V01.00.xml |
| BCG_Package_RNIFV02.00_3C7V01.00.zip | BCG_RNIFV02.00_3C7V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3C7V01.00.zip | BCG_RNSC1.0_RNIF1.1_3C7V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3C7V01.00.zip | BCG_RNSC1.0_RNIFV02.00_3C7V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3C7 Notify of Self-Billing Invoice:

- BCG_3C7SelfBillingInvoiceNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V422.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalDocumentTypeCode.xsd
- BCG_GlobalDocumentTypeCode_V422.xsd

- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPaymentTermsCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalSaleTypeCode.xsd
- BCG_GlobalShipmentTermsCode.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_InvoiceChargeTypeCode.xsd
- BCG_NationalExportControlClassificationCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 3D8 Distribute Work in Process

The following section describes the contents for the 3D8 Distribute Work in Process PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 3D8 Distribute Work in Process PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 82. 3D8 Distribute Work in Process ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_3D8V01.00.zip | BCG_RNIF1.1_3D8V01.00.xml |
| BCG_Package_RNIFV02.00_3D8V01.00.zip | BCG_RNIFV02.00_3D8V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_3D8V01.00.zip | BCG_RNSC1.0_RNIF1.1_3D8V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_3D8V01.00.zip | BCG_RNSC1.0_RNIFV02.00_3D8V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 3D8 Distribute Work in Process:

- BCG_3D8WorkInProcessNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalLotCode.xsd
- BCG_GlobalLotQuantityClassificationCode.xsd
- BCG_GlobalLotStatusCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPriorityCode.xsd

- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalWorkInProcessLocationCode.xsd
- BCG_GlobalWorkInProcessPartTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 4A1 Notify of Strategic Forecast

The following section describes the contents for the 4A1 Notify of Strategic Forecast PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 4A1 Notify of Strategic Forecast PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 83. 4A1 Notify of Strategic Forecast ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4A1V02.00.zip | BCG_RNIF1.1_4A1V02.00.xml |
| BCG_Package_RNIFV02.00_4A1V02.00.zip | BCG_RNIFV02.00_4A1V02.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_4A1V02.00.zip | BCG_RNSC1.0_RNIF1.1_4A1V02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4A1V02.00.zip | BCG_RNSC1.0_RNIFV02.00_4A1V02.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 4A1 Notify of Strategic Forecast:

- BCG_4A1StrategicForecastNotification_V02.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalForecastEventCode.xsd
- BCG_GlobalForecastIntervalCode.xsd
- BCG_GlobalForecastTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerReferenceTypeCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_StrategicForecastQuantityTypeCode.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

# 4A3 Notify of Threshold Release Forecast

The following section describes the contents for the 4A3 Notify of Threshold Release Forecast PIP.

## Package file contents

The following table shows the ZIP files and corresponding XML files for the 4A3 Notify of Threshold Release Forecast PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 84. 4A3 Notify of Threshold Release Forecast ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4A3V02.00.zip | BCG_RNIF1.1_4A3V02.00.xml |
| BCG_Package_RNIFV02.00_4A3V02.00.zip | BCG_RNIFV02.00_4A3V02.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_4A3V02.00.zip | BCG_RNSC1.0_RNIF1.1_4A3V02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4A3V02.00.zip | BCG_RNSC1.0_RNIFV02.00_4A3V02.00.xml |

## Guideline map contents

This section lists the guideline maps contents for 4A3 Notify of Threshold Release Forecast:

- BCG_4A3ThresholdReleaseForecastNotification_V02.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalForecastEventCode.xsd
- BCG_GlobalForecastIntervalCode.xsd
- BCG_GlobalForecastInventoryTypeCode.xsd
- BCG_GlobalForecastReferenceTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerReferenceTypeCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_OrderForecastQuantityTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

# 4A4 Notify of Planning Release Forecast

The following section describes the contents for the 4A4 Notify of Planning Release Forecast PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 4A4 Notify of Planning Release Forecast. PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 85. 4A4 Notify of Planning Release Forecast ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4A4R02.00A.zip | BCG_RNIF1.1_4A4R02.00A.xml |
| BCG_Package_RNIFV02.00_4A4R02.00A.zip | BCG_RNIFV02.00_4A4R02.00A.xml |
| BCG_Package_RNSC1.0_RNIF1.1_4A4R02.00A.zip | BCG_RNSC1.0_RNIF1.1_4A4R02.00A.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4A4R02.00A.zip | BCG_RNSC1.0_RNIFV02.00_4A4R02.00A.xml |

### Guideline map contents

This section lists the guideline maps contents for 4A4 Notify of Planning Release Forecast:

- BCG_4A4PlanningReleaseForecastNotification_R02.00A.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalForecastInventoryTypeCode.xsd
- BCG_GlobalForecastQuantityTypeCode_V422.xsd
- BCG_GlobalForecastReferenceTypeCode.xsd
- BCG_GlobalIntervalCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerReferenceTypeCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalTransportEventCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_PhysicalAddress_Types_V422.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 4A5 Notify of Forecast Reply

The following section describes the contents for the 4A5 Notify of Forecast Reply PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 4A5 Notify of Forecast Reply PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 86. 4A5 Notify of Forecast Reply ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4A5V02.00.zip | BCG_RNIF1.1_4A5V02.00.xml |
| BCG_Package_RNIFV02.00_4A5V02.00.zip | BCG_RNIFV02.00_4A5V02.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_34A5V02.00.zip | BCG_RNSC1.0_RNIF1.1_34A5V02.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4A5V02.00.zip | BCG_RNSC1.0_RNIFV02.00_4A5V02.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 4A5 Notify of Forecast Reply:

- BCG_4A5ForecastReplyNotification_V02.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ForecastReplyQuantityTypeCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalForecastEventCode.xsd
- BCG_GlobalForecastIntervalCode.xsd
- BCG_GlobalForecastInventoryTypeCode.xsd
- BCG_GlobalForecastReferenceTypeCode.xsd
- BCG_GlobalForecastResponseCode.xsd
- BCG_GlobalForecastRevisionReasonCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerReferenceTypeCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 4B2 Notify of Shipment Receipt

The following section describes the contents for the 4B2 Notify of Shipment Receipt PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 4B2 Notify of Shipment Receipt PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 87. 4B2 Notify of Shipment Receipt ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4B2V01.00.zip | BCG_RNIF1.1_4B2V01.00.xml |
| BCG_Package_RNIFV02.00_4B2V01.00.zip | BCG_RNIFV02.00_4B2V01.00.xml |

*Table 87. 4B2 Notify of Shipment Receipt ZIP and XML files (continued)*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNSC1.0_RNIF1.1_4B2V01.00.zip | BCG_RNSC1.0_RNIF1.1_4B2V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4B2V01.00.zip | BCG_RNSC1.0_RNIFV02.00_4B2V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 4B2 Notify of Shipment Receipt:

- BCG_4B2ShipmentReceiptNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalLotDiscrepancyReasonCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalReceivingDiscrepancyCode.xsd
- BCG_GlobalReceivingDiscrepancyReasonCode.xsd
- BCG_GlobalSpecialFulfillmentRequestCode.xsd
- BCG_GlobalSpecialHandlingCode.xsd
- BCG_GlobalTrackingReferenceTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 4B3 Notify of Consumption

The following section describes the contents for the 4B3 Notify of Consumption PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 4B3 Notify of Consumption PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 88. 4B3 Notify of Consumption ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4B3V01.00.zip | BCG_RNIF1.1_4B3V01.00.xml |
| BCG_Package_RNIFV02.00_4B3V01.00.zip | BCG_RNIFV02.00_4B3V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_4B3V01.00.zip | BCG_RNSC1.0_RNIF1.1_4B3V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4B3V01.00.zip | BCG_RNSC1.0_RNIFV02.00_4B3V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 4B3 Notify of Consumption:

- BCG_4B3ConsumptionNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessDescription_Types_V422.xsd
- BCG_BusinessDescription_Types_V43.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V43.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ContactInformation_Types_V422.xsd
- BCG_ContactInformation_Types_V43.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode_V43.xsd
- BCG_GlobalInventoryCode_V43.xsd
- BCG_GlobalPartnerClassificationCode_V422.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 4C1 Distribute Inventory Report V02.01

The following section describes the contents for the 4C1 Distribute Inventory Report V02.01PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 4C1 Distribute Inventory Report PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 89. 4C1 Distribute Inventory Report ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4C1V02.01.zip | BCG_RNIF1.1_4C1V02.01.xml |
| BCG_Package_RNIFV02.00_4C1V02.01.zip | BCG_RNIFV02.00_4C1V02.01.xml |
| BCG_Package_RNSC1.0_RNIF1.1_4C1V02.01.zip | BCG_RNSC1.0_RNIF1.1_4C1V02.01.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4C1V02.01.zip | BCG_RNSC1.0_RNIFV02.00_4C1V02.01.xml |

### Guideline map contents

This section lists the guideline maps contents for 4C1 Distribute Inventory Report:

- BCG_4C1InventoryReportNotification_V02.01.xsd
- BCG_BusinessDescription_Types.xsd

- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ContactInformation_Types_V422.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalInventoryCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_PhysicalAddress_Types_V422.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 4C1 Distribute Inventory Report V02.03

The following section describes the contents for the 4C1 Distribute Inventory Report V02.03 PIP:

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 4C1 Distribute Inventory Report PIP. The guideline maps, which are common to all versions, are shown in the section that follows:

*Table 90. 4C1 Distribute Inventory Report ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_4C1V02.03.zip | BCG_RNIF1.1_4C1V02.03.xml |
| BCG_Package_RNIFV02.00_4C1V02.03.zip | BCG_RNIFV02.00_4C1V02.03.xml |
| BCG_Package_RNSC1.0_RNIF1.1_4C1V02.03.zip | BCG_RNSC1.0_RNIF1.1_4C1V02.03.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_4C1V02.03.zip | BCG_RNSC1.0_RNIFV02.00_4C1V02.03.xml |

### Guideline map contents

This section lists the guideline maps contents for 4C1 Distribute Inventory Report:

- BCG_4C1InventoryReportNotification_V02.03.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalInventoryCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd

- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 5C1 Distribute Product List

The following section describes the contents for the 5C1 Distribute Product List PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 5C1 Distribute Product List PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 91. 5C1 Distribute Product List ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_5C1V01.00.zip | BCG_RNIF1.1_5C1V01.00.xml |
| BCG_Package_RNIFV02.00_5C1V01.00.zip | BCG_RNIFV02.00_5C1V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_5C1V01.00.zip | BCG_RNSC1.0_RNIF1.1_5C1V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_5C1V01.00.zip | BCG_RNSC1.0_RNIFV02.00_5C1V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 5C1 Distribute Product List:

- BCG_5C1ProductListNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPriceTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 5C2 Request Design Registration

The following section describes the contents for the 5C2 Request Design Registration PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 5C2 Request Design Registration PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 92. 5C2 Request Design Registration ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_5C2V01.00.zip | BCG_RNIF1.1_5C2V01.00.xml |
| BCG_Package_RNIFV02.00_5C2V01.00.zip | BCG_RNIFV02.00_5C2V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_5C2V01.00.zip | BCG_RNSC1.0_RNIF1.1_5C2V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_5C2V01.00.zip | BCG_RNSC1.0_RNIFV02.00_5C2V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 5C2 Request Design Registration:

- BCG_5C2DesignRegistrationConfirmation_V01.00.xsd
- BCG_5C2DesignRegistrationRequest_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_common.xsd
- BCG_common_V422.xsd
- BCG_common_V43.xsd
- BCG_ContactInformation_Types.xsd
- BCG_DesignWinStatusReasonCode_V43.xsd
- BCG_GlobalAttachmentDescriptionCode_V422.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalMimeTypeQualifierCode_V43.xsd
- BCG_GlobalMonetaryAmountTypeCode.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd
- BCG_GlobalPartnerRoleClassificationCode_V43.xsd
- BCG_GlobalPriceTypeCode_V43.xsd
- BCG_GlobalRegistrationComplexityLevelCode.xsd
- BCG_GlobalRegistrationInvolvementLevelCode.xsd
- BCG_InvoiceChargeTypeCode_V422.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 5C4 Distribute Registration Status

The following section describes the contents for the 5C4 Distribute Registration Status PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 5C4 Distribute Registration Status PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 93. 5C4 Distribute Registration Status ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_5C4V01.02.zip | BCG_RNIF1.1_5C4V01.02.xml |
| BCG_Package_RNIFV02.00_5C4V01.02.zip | BCG_RNIFV02.00_5C4V01.02.xml |
| BCG_Package_RNSC1.0_RNIF1.1_5C4V01.023.zip | BCG_RNSC1.0_RNIF1.1_5C4V01.023.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_5C4V01.02.zip | BCG_RNSC1.0_RNIFV02.00_5C4V01.02.xml |

### Guideline map contents

This section lists the guideline maps contents for 5C4 Distribute Registration Status:

- BCG_5C4RegistrationStatusNotification_V01.02.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalRegistrationComplexityLevelCode.xsd
- BCG_GlobalRegistrationInvolvementLevelCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 5D1 Request Ship From Stock And Debit Authorization

The following section describes the contents for the 5D1 Request Ship From Stock And Debit Authorization PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 5D1 Request Ship From Stock And Debit Authorization PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 94. 5D1 Request Ship from Stock and Debit Authorization ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_5D1V01.00.zip | BCG_RNIF1.1_5D1V01.00.xml |
| BCG_Package_RNIFV02.00_5D1V01.00.zip | BCG_RNIFV02.00_5D1V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_5D1V01.00.zip | BCG_RNSC1.0_RNIF1.1_5D1V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_5D1V01.00.zip | BCG_RNSC1.0_RNIFV02.00_5D1V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 5D1 Request Ship From Stock And Debit Authorization:

- BCG_5D1ShipFromStockAndDebitAuthorizationConfirmation_V01.00.xsd
- BCG_5D1ShipFromStockAndDebitAuthorizationRequest_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPriceTypeCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalShipFromStockAndDebitAuthorizationRejectionCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 6C1 Query Service Entitlement

The following section describes the contents for the 6C1 Query Service Entitlement PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 6C1 Query Service Entitlement PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 95. 6C1 Query Service Entitlement ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_6C1V01.00.zip | BCG_RNIF1.1_6C1V01.00.xml |
| BCG_Package_RNIFV02.00_6C1V01.00.zip | BCG_RNIFV02.00_6C1V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_6C1V01.00.zip | BCG_RNSC1.0_RNIF1.1_6C1V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_6C1V01.00.zip | BCG_RNSC1.0_RNIFV02.00_6C1V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 6C1 Query Service Entitlement:

- BCG_6C1ServiceEntitlementQuery_V01.00.xsd
- BCG_6C1ServiceEntitlementStatusResponse_V01.00.xsd
- BCG_common_V43.xsd
- BCG_ContactInformation_Types.xsd
- BCG_ContactInformation_Types_V43.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalNotificationCode_V43.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd

- BCG_GlobalPaymentTypeCode_V43.xsd
- BCG_GlobalServiceDeliveryMethodCode_V43.xsd
- BCG_GlobalShippingServiceLevelCode.xsd
- BCG_GlobalWarrantyMethodCode_V43.xsd
- BCG_GlobalWarrantyProgramCode_V43.xsd
- BCG_GlobalWarrantyTypeCode_V43.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

## 6C2 Request Warranty Claim

The following section describes the contents for the 6C2 Request Warranty Claim PIP.

### Package file contents

The following table shows the ZIP files and corresponding XML files for the 6C2 Request Warranty Claim PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 96. 6C2 Request Warranty Claim ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_6C2V01.00.zip | BCG_RNIF1.1_6C2V01.00.xml |
| BCG_Package_RNIFV02.00_6C2V01.00.zip | BCG_RNIFV02.00_6C2V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_6C2V01.00.zip | BCG_RNSC1.0_RNIF1.1_6C2V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_6C2V01.00.zip | BCG_RNSC1.0_RNIFV02.00_6C2V01.00.xml |

### Guideline map contents

This section lists the guideline maps contents for 6C2 Request Warranty Claim:

- BCG_6C2WarrantyClaimConfirmation_V01.00.xsd
- BCG_6CWarrantyClaimRequest_V01.00.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalCurrencyCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalFailureTypeCode_V43.xsd
- BCG_GlobalOperatingSystemCode_V43.xsd
- BCG_GlobalPartnerClassificationCode_V43.xsd
- BCG_GlobalPartnerRoleClassificationCode_V43.xsd
- BCG_GlobalPaymentTypeCode_V43.xsd
- BCG_GlobalServiceDeliveryMethodCode_V43.xsd
- BCG_GlobalWarrantyTypeCode_V43.xsd
- BCG_PartnerDescription_Types_V43.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

# 7B1 Distribute Work in Process

The following section describes the contents for the 7B1 Distribute Work in Process PIP.

## Package file contents

The following table shows the ZIP files and corresponding XML files for the 7B1 Distribute Work in Process PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 97. 7B1 Distribute Work in Process ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_7B1V01.00.zip | BCG_RNIF1.1_7B1V01.00.xml |
| BCG_Package_RNIFV02.00_37B1V01.00.zip | BCG_RNIFV02.00_37B1V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_7B1V01.00.zip | BCG_RNSC1.0_RNIF1.1_7B1V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_7B1V01.00.zip | BCG_RNSC1.0_RNIFV02.00_7B1V01.00.xml |

## Guideline map contents

This section lists the guideline maps contents for 7B1 Distribute Work in Process:

- BCG_7B1WorkInProcessNotification_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalChangeReasonCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalEquipmentTypeCode.xsd
- BCG_GlobalLotCode.xsd
- BCG_GlobalLotQuantityClassificationCode.xsd
- BCG_GlobalLotStatusCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPriorityCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalWorkInProcessLocationCode.xsd
- BCG_GlobalWorkInProcessPartTypeCode.xsd
- BCG_GlobalWorkInProcessQuantityChangeCode.xsd
- BCG_GlobalWorkInProcessTypeCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

# 7B5 Notify Of Manufacturing Work Order

The following section describes the contents for the 7B5 Notify Of Manufacturing Work Order PIP.

## Package file contents

The following table shows the ZIP files and corresponding XML files for the 7B5 Notify Of Manufacturing Work Order PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 98. 7B5 Notify of Manufacturing Work Order ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_7B5V01.00.zip | BCG_RNIF1.1_7B5V01.00.xml |
| BCG_Package_RNIFV02.00_7B5V01.00.zip | BCG_RNIFV02.00_7B5V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_7B5V01.00.zip | BCG_RNSC1.0_RNIF1.1_7B5V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_7B5V01.00.zip | BCG_RNSC1.0_RNIFV02.00_7B5V01.00.xml |

## Guideline map contents

This section lists the guideline maps contents for 7B5 Notify Of Manufacturing Work Order:

- BCG_7B5NotifyOfManufacturingWorkOrder_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalAttachmentDescriptionCode_V422.xsd
- BCG_GlobalBusinessActionCode_V422.xsd
- BCG_GlobalChangeReasonCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDevicePackageTypeCode_V422.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalLineItemStatusCode.xsd
- BCG_GlobalLotCode.xsd
- BCG_GlobalMimeTypeQualifierCode_V422.xsd
- BCG_GlobalPackageTypeCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalPhysicalUnitOfMeasureCode.xsd
- BCG_GlobalPriorityCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_GlobalWorkInProcessLocationCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

# 7B6 Notify Of Manufacturing Work Order Reply

The following section describes the contents for the 7B6 Notify Of Manufacturing Work Order Reply PIP.

## Package file contents

The following table shows the ZIP files and corresponding XML files for the 7B6 Notify Of Manufacturing Work Order Reply PIP. The guideline maps, which are common to all versions, are shown in the section that follows.

*Table 99. 7B6 Notify of Manufacturing Work Order Reply ZIP and XML files*

| ZIP file name | XML file name |
|---|---|
| BCG_Package_RNIF1.1_7B6V01.00.zip | BCG_RNIF1.1_7B6V01.00.xml |
| BCG_Package_RNIFV02.00_7B6V01.00.zip | BCG_RNIFV02.00_7B6V01.00.xml |
| BCG_Package_RNSC1.0_RNIF1.1_7B6V01.00.zip | BCG_RNSC1.0_RNIF1.1_7B6V01.00.xml |
| BCG_Package_RNSC1.0_RNIFV02.00_7B6V01.00.zip | BCG_RNSC1.0_RNIFV02.00_7B6V01.00.xml |

## Guideline map contents

This section lists the guideline maps contents for 7B6 Notify Of Manufacturing Work Order Reply:

- BCG_7B6NotifyOfManufacturingWorkOrderReply_V01.00.xsd
- BCG_BusinessDescription_Types.xsd
- BCG_BusinessTaxIdentifier_Types.xsd
- BCG_common.xsd
- BCG_ContactInformation_Types.xsd
- BCG_GlobalChangeReasonCode.xsd
- BCG_GlobalCountryCode.xsd
- BCG_GlobalDocumentReferenceTypeCode.xsd
- BCG_GlobalLineItemStatusCode.xsd
- BCG_GlobalPartnerClassificationCode.xsd
- BCG_GlobalPartnerRoleClassificationCode.xsd
- BCG_GlobalProductUnitOfMeasureCode.xsd
- BCG_PartnerDescription_Types.xsd
- BCG_PhysicalAddress_Types.xsd
- BCG_string_len_0.xsd
- BCG_xml.xsd

# Chapter 22. Additional CIDX information

This appendix gives you additional information about CIDX support. It includes the following topics:

**Related reference**

"CIDX process enablement support"

"Creating CIDX document definition packages"

## CIDX process enablement support

CIDX provides following two mechanisms for process enablement:

- **Message-based enablement:** document linkage is based on <RequestingDocumentIdentifier> and <ThisDocumentIdentifier>
- **Framework-based enablement:** document linkage is based on RNIF 1.1 service header semantics

For message-based enablement, 1-action PIP packages for ChemXML transactions are required. Whereas for framework-based enablement, 2-action PIP packages for ChemXML transactions are required. WebSphere Partner Gateway supports both the forms of process enablement. WebSphere Partner Gateway provides the 1-action PIP packages for "E41 Order Create" and "E42 Order Response".

## Creating CIDX document definition packages

You might need to create your own CIDX packages to support other CIDX messages. The procedure to create new CIDX document definition packages is the same as that for RosettaNet.

For additional information on RosettaNet, see Chapter 21, "Additional RosettaNet information," on page 351

# Chapter 23. Attributes

This appendix describes attributes you can set from the Community Console. The following attributes are described:



## EDI attributes

This section provides a description of the EDI attributes that you can use while setting up your EDI exchanges. Some of these attributes are predefined in the control string representing the transformation map associated with the EDI document. The values set in the control string (at the Data Interchange Services client) override any values you enter at the Community Console.

### Envelope profile attributes

You can set various attributes for an EDI envelope profile. The attributes that are available depend on the EDI Type. In general, the attributes correspond to an EDI standard, and the allowable values depend on the EDI standard the envelope profile represents.

None of the attributes requires a value. For some of the attributes, a default value is used if you do not enter a value. The tables in this section list the attributes that have associated defaults and their default values.

**Note:** The envelope profile properties not listed do not have default values. The text value you specify is used if it is not overridden by generic or specific envelope properties set in the map or in a connection.

#### X12 attributes

The tables in this section list the X12 attributes for which default values are supplied.

##### General attributes

Table 100 lists the General attributes for which default values are provided.

*Table 100. General attributes*

| Field name | Required? | Description | Default |
|---|---|---|---|
| INTCTLLEN (Interchange Control Number Length) | No | Defines a specific length for the interchange control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |

*Table 100. General attributes (continued)*

| Field name | Required? | Description | Default |
|---|---|---|---|
| GRPCTLLEN (Group Control Number Length) | No | Defines a specific length for the group control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |
| TRXCTLLEN (Transaction Control Number Length) | No | Defines a specific length for the transaction control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |
| ENVTYPE (Envelope Type) | No | This attribute is not set by the user but is derived from the envelope profile type being created. | X12 |
| MAXDOCS (Max Transactions Number) | No | Maximum number of transactions in an envelope. If you enter a value, it must be an integer. | No maximum |
| CTLNUMFLAG (Control Numbers by Transaction ID) | No | Yes indicates that separate sets of control numbers are kept based on the EDI transaction type.<br><br>No indicates that a common set of control numbers for any EDI transaction type should be used. | No |

### Interchange attributes

No X12 interchange attributes are required, and the attributes do not have default values.

*Table 101. Group attributes*

| Field name | Required? | Description | Default |
|---|---|---|---|
| GS01 (Functional group ID) | No | The group identifier. | The default value comes from the control-string header. You can view this value in the Data Interchange Services client by looking at the Functional Group column of the EDI Document Definitions page. |
| GS08 (Group version) | No | The group version. | The default value is per the standard. |

### Group attributes

Table 101 lists the group attributes for which default values are provided.

### Transaction attributes

No transaction attributes are required. The attributes do not have default values.

## UCS attributes

This section lists whether default values apply to a UCS interchange, group, and transaction.

### General attributes

Table 102 on page 411 lists the General attributes for which default values are provided.

*Table 102. General attributes*

| Field name | Required? | Description | Default |
|---|---|---|---|
| INTCTLLEN (Interchange Control Number Length) | No | Defines a specific length for the interchange control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 5 |
| GRPCTLLEN (Group Control Number Length) | No | Defines a specific length for the group control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |
| TRXCTLLEN (Transaction Control Number Length) | No | Defines a specific length for the transaction control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |
| ENVTYPE (Envelope Type) | No | This attribute is not set by the Hub Admin but is derived from the envelope profile type being created. | UCS |
| MAXDOCS (Max Transactions Number) | No | Maximum number of transactions in an envelope. If you enter a value, it must be an integer. | No maximum |
| CTLNUMFLAG (Control Numbers by Transaction ID) | No | Yes indicates that separate sets of control numbers are kept based on the EDI transaction type.<br><br>No indicates that a common set of control numbers for any EDI transaction type should be used. | No |

### Interchange attributes

No interchange attributes are required. The attributes do not have default values.

### Group attributes

Table 103 lists the group attributes for which default values are provided.

*Table 103. Group attributes*

| Field name | Required? | Description | Default |
|---|---|---|---|
| GS01 (Functional group ID) | No | The group identifier. | The default value comes from the control-string header. You can view this value in the Data Interchange Services client by looking at the Functional Group column of the EDI Document Definitions page. |
| GS08 (Group version) | No | The group version. | The default value is per the standard. |

### Transaction attributes

No transaction attributes are required. The attributes do not have default values.

## EDIFACT attributes

This section lists whether default values apply to an EDIFACT interchange, group, and message.

### General attributes

Table 104 lists the General attributes for which default values are provided.

*Table 104. General attributes*

| Field name | Required? | Description | Default |
|---|---|---|---|
| INTCTLLEN (Interchange Control Number Length) | No | Defines a specific length for the interchange control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |
| GRPCTLLEN (Group Control Number Length) | No | Defines a specific length for the group control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |
| TRXCTLLEN (Transaction Control Number Length) | No | Defines a specific length for the transaction control number. If you enter a value, it must be an integer.<br><br>If no value is entered, the default length is used. | 9 |
| ENVTYPE (Envelope Type) | No | This attribute is not set by the Hub Admin but is derived from the envelope profile type being created. | EDIFACT |
| EDIFACTGRP (Create Groups for EDI) | No | This value is only for EDIFACT envelope types. (The group level has been deprecated in EDIFACT.)<br><br>Yes indicates that functional groups (UNG/UNE segments) should be created for EDIFACT DATA.<br><br>No indicates that they should not. | No |
| MAXDOCS (Max Transactions Number) | No | Maximum number of transactions in an envelope. If you enter a value, it must be an integer. | No maximum |
| CTLNUMFLAG (Control Numbers by Transaction ID) | No | Yes indicates that separate sets of control numbers are kept based on the EDI transaction type.<br><br>No indicates that a common set of control numbers for any EDI transaction type should be used. | No |

### Interchange attributes

No interchange attributes are required. The attributes do not have default values.

### Group attributes

Table 105 lists the group attributes for which default values are provided.

*Table 105. Group attributes*

| Field name | Required? | Description | Default |
|---|---|---|---|
| UNG01 (Functional group ID) | No | The group identifier. | The default value comes from the control-string header. You can view this value in the Data Interchange Services client by looking at the Functional Group column of the EDI Document Definitions page. |

### Message attributes

Table 106 on page 413 lists the message attributes for which default values are provided.

*Table 106. Message attributes*

| Field name | Required? | Description | Default |
|---|---|---|---|
| UNH0201 (Message Type) | No | The type of message. | The default value comes from the control-string header. You can view this value in the Data Interchange Services client by looking at the EDI Document Definitions page. |
| UNH0202 (Message Version) | No | The version of the message. | D |
| UNH0203 (Message Release) | No | The release of the message. | Per the standard |
| UNH0204 (Controlling Agency) | No | The code identifying a controlling agency. | UN |

# Document definition and connection attributes

This section lists document definition attributes for the envelope. Some of these attributes can be set only at the protocol or connection level, as indicated.

## Separator and delimiter attributes

This section lists the characters used as delimiters or separators within an EDI interchange. Table 107 shows the attribute as it appears on the Community Console, the corresponding term in X12 and EDIFACT (ISO 9735 Version 4, Release 1), whether the attribute is required, and a description of the attribute. Following the table is an example of how these characters appear in an EDI document.

### Attribute descriptions

The separator and delimiter attributes are listed in Table 107.

**Note:** Some characters (as noted) can be hexadecimal values. These can be Unicode values or values from another type of encoding. For Unicode, use the format \unnnn. For other encoding, use the form 0xnn.

*Table 107. Envelope profile attributes*

| Attribute | X12 term | EDIFACT term | Description |
|---|---|---|---|
| Segment delimiter | segment terminator | segment terminator | This is a single character, which appears at the last character of a segment. The character can be a hexadecimal value.<br><br>The default value is based on the EDI type.<br><br>**X12** ~ (tilde)<br><br>**EDIFACT** ‘ (single quote)<br><br>**UCS** ~ (tilde) |

*Table 107. Envelope profile attributes (continued)*

| Attribute | X12 term | EDIFACT term | Description |
|---|---|---|---|
| Data element delimiter | data element separator | data element separator | This is a single character, which separates the data elements of a segment. The character can be a hexadecimal value.<br><br>The default value is the based on the EDI type.<br><br>**X12** * (asterisk)<br><br>**EDIFACT**<br>+ (plus sign)<br><br>**UCS** * (asterisk) |
| Subelement delimiter | component element separator | component data element separator | This is a single character, which separates the component elements of a composite data element. The character can be a hexadecimal value.<br><br>The default value is the based on the EDI type.<br><br>**X12** \ (back slash)<br><br>**EDIFACT**<br>: (colon)<br><br>**UCS** \ (back slash) |
| Release character | | release character | This is a single character, which overrides the meaning of the next character, allowing a separator character to appear within a data element. The character can be a hexadecimal value. It applies to EDIFACT only.<br><br>**EDIFACT**<br>? (question mark) |
| Repeating data element separator | repetition separator | repetition separator | This is a single character, which separates the instances of a repeating data element. This character can be a hexadecimal value.<br><br>The default value is based on the EDI type for X12 or EDIFACT.<br><br>**X12** ^ (hat sign, accent circumflex)<br><br>**EDIFACT**<br>* (asterisk) |
| Decimal notation | | decimal notation (deprecated) | This attribute was used in decimal formatting or parsing and is now deprecated. It can be a period or comma only.<br><br>The default value is a period. |

## Example EDI structure

This section shows a simple EDI interchange and how the attributes described in Table 107 on page 413 are used in an interchange.

An EDI message consists of a series of segments in a particular order. A segment consists of a series of elements. In a segment, an element can be a simple data element, which contains only one item of information. An element can also be a composite data element, containing two or more simple data elements. The simple elements that make up a composite element are called component data elements.

There is no nesting of composite data elements. A composite element can contain only simple data elements, not other composites. Although not shown here, a component data element can also be defined as a repeating data element.

Consider the following example:

```
ABC*123*AA\BB\CC*001^002^003*star?*power~
```

In this example:

- "ABC" is the segment name (EDIFACT calls this the "segment tag"); this would be called an "ABC segment"
- "*" (asterisk) is the data element separator.

  The corresponding attribute name on the Community Console is Segment delimiter.
- "123" is the first data element, a simple data element (which might be referred to as ABC01 is some contexts)
- "AA\BB\CC" is the second data element (ABC02), a composite element made up of component data elements
  - "\ " (backslash) is the component data element separator

    The corresponding attribute name on the Community Console is the Data element delimiter.
  - "AA" is the first component data element of ABC02 (which might be designated ABC0201)
  - "BB" is the second component data element of ABC02 (ABC0202)
  - "CC" is the third component data element of ABC02 (ABC0203)
- "001^002^003" is the third data element (ABC03), a repeating data element
  - "^" (hat sign) is the repetition separator

    The corresponding attribute name on the Community Console is the Repeating data element character.
  - "001","002", "003" are the repetitions (all would be designated ABC03)
- "star?*power" is the fourth data element (ABC04)
  - "?" (question mark) is the release character, meaning the following asterisk is not treated as a data element separator
  - "star*power" is the resulting value of ABC04
- "~" (tilde) is the segment terminator.

  The corresponding attribute name on the Community Console is Segment delimiter.

## Additional EDI attributes

This section lists additional EDI attributes that you can set at the document definition level or the connection level.

*Table 108. Additional EDI attributes*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Segment output | No | Used in EDI/XML transformation, this indicates whether a line break should occur after each EDI segment or XML element.<br><br>**Important:**<br>1. Always use a single character delimiter.<br>2. If you use combination of "/r/n" character delimiter, and if "/r" character delimiter is found at the segment delimiter position of Interchange Header, then the "/n" character delimiter will be ignored.<br>3. Modify the type tree accordingly. | Limited to protocol or connection | Yes |
| Allow documents with duplicate document IDs | No | Yes indicates that duplicate document IDs (interchange control numbers) are allowed.<br><br>No indicates that duplicate interchange control numbers should be treated as an error. | Limited to protocol or connection | No |
| Max error level at Transformation | No | Indicates the maximum number of errors that can occur during a transformation before the transformation fails.<br><br>Valid values are 0, 1, or 2.<br><br>If the transformation map contains an Error command to indicate a user-specified error, and the level parameter of the Error command is greater than this value, the transformation fails. | Limited to protocol or connection | 0 |
| FA Map | No | Provides the map to use for converting the internal generic FA to the specific FA.<br>**Note:** You select this attribute from a list of maps identified as FA maps (map type of “K”). | Limited to protocol or connection | |
| Envelope Profile | Yes | The EDI envelope profile name to use for enveloping. All envelope profiles that you have defined are available from the list. | | |
| XMLNS Active | No | Do namespace processing for the input XML document. This attribute is used by the XML transformation step.<br><br>Valid values are Yes or No. | | Schema: Yes<br>DTD: No |
| Max validation error level | No | The maximum acceptable validation error level (the error severity to accept before considering the transaction "failed").<br><br>Valid values are 0, 1, or 2.<br><br>**0** Allow only validation with no errors.<br><br>**1** Do not fail documents that have only simple element validation errors.<br><br>**2** Do not fail documents that have element or segment validation errors. | | 0 |

*Table 108. Additional EDI attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Validation level | No | Indicates the level of checking to be performed at the transaction level. A value of 2 means to use the values set for the Alphanumeric validation table and Char Set validation table attributes. This attribute also applies to the Detailed validation of segments attribute if that attribute is set to Yes.<br><br>Valid values are 0, 1, or 2.<br><br>**0** Only perform basic validation, such as checking for missing mandatory elements and segments and minimum or maximum lengths. Do not validate element values against the data types or code lists specified in the transaction definition.<br><br>**1** Perform level 0 validation, plus validate the element values against the code lists specified for the data element.<br><br>**2** Perform level 1 validation, plus validate that the element value is correct for the data type of the element. | | 0 |
| Char set validation table | No | Indicates the table to use for character set validation. This table is used only when the Validation level attribute is 2.<br><br>This attribute refers to the virtual code lists table. The user can create new code lists in the Code Lists tab of the Mapping area in the Data Interchange Services client. This area also contains code lists that are used for other purposes, such as validation of certain EDI elements. | | CHARSET |
| Alphanumeric validation table | No | Indicates the table to use for alphanumeric validation. This table is used only when the Validation level attribute is 2.<br><br>The attribute refers to the virtual code list tables. The user can create new code lists in the Code Lists tab of the Mapping area in the Data Interchange Services client. This area also contains code lists that are used for other purposes, such as validation of certain EDI elements. | | ALPHANUM |
| Generate group level info only in functional Ack | No | This attribute applies to EDI-X12. The values are Yes or No.<br><br>**Yes** Generate group level information only for functional acknowledgment.<br><br>**No** Generate full functional acknowledgment detail (for each individual transaction and segments and elements within a transaction). | Limited to protocol or connection | No |

*Table 108. Additional EDI attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Century control year | No | When dates are being converted from two-digit years to four-digit years, two-digit years after this value are assumed to have a century value of "19". Two-digit years equal to or before this value are assumed to have a century value of "20".<br><br>The valid range is 0-99. | Limited to protocol or connection | 10 |
| Detailed validation of segment | No | This attribute applies to the following segment headers and trailers:<br>• X12<br>– ISA, IEA<br>– GS, GE<br>– ST, SE<br>• EDIFACT<br>– UNA<br>– UNB, UNZ<br>– UNG, UNE<br>– UNH, UNT<br>• UNTUCS<br>– BG, EG<br>– GS, GE<br>– ST, SE<br><br>Valid values are Yes or No.<br><br>**Yes** Perform detailed envelope segment validation. The depth of checking is controlled by the Validation level attribute.<br><br>**No** Do not perform detailed envelope segment validation. | Limited to protocol or connection | No |
| TA1 override | No | Allow generation of a TA1 request if indicated in the Interchange envelope segment. Applies only to EDI-X12.<br><br>If set to Yes, a TA1 is generated if specified in the Interchange envelope segment.<br><br>If set to No, a TA1 is not generated, even if it was specified in the Interchange envelope segment. | Limited to protocol or connection | Yes |
| Discard on error | No | This attribute is used in polymorphic processing.<br><br>In the case of a batch that results from de-enveloping, this attribute indicates whether to discard the entire batch if any of the transactions fail.<br><br>Valid values are Yes and No. | Limited to protocol or connection | No |

*Table 108. Additional EDI attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Connection Profile Qualifier1 | No | This attribute is used by the Enveloper to determine which profile to use for an interchange connection. Transactions with different values for this attribute are put into different interchanges. | | |
| Interchange qualifier | No | The code used to identify the format of the interchange sender or receiver identifier. | | |
| Interchange identifier | No | Identifies the specific sender or receiver of the document. The type of data entered is determined by the Interchange qualifier attribute. | | |
| Interchange usage indicator | No | Indicates whether the source documents being translated are classified as Production, Test, or Information documents.<br><br>Valid values are P, T, and I. | | |
| Group application sender identifier | No | Identifies the specific sender of the transaction. This attribute, when agreed to by trading partners, facilitates addressing within a company. | | |
| Group application receiver identifier | No | Identifies the specific receiver or application of the transaction. This attribute, when agreed to by trading partners, facilitates addressing within a company. | | |
| Interchange reverse routing | No | Indicates the address to which the recipient should address any replies. | | |
| Interchange routing address | No | The sub-address code for onward routing. | | |
| Group application sender qualifier | No | The code used to identify the format of the group application sender identifier. | | |
| Group application receiver qualifier | No | The code used to identify the format of the group application receiver identifier. | | |
| Group application password | No | This attribute defines security information. | | |
| FA required time limit | | Number of minutes after a transaction is sent in which an FA is required to return. If the value is blank, no FA is required. | | |

## Data Interchange Services client properties

This section lists the properties that can be set as part of the transformation map in the Data Interchange Services client and their corresponding WebSphere Partner Gateway attributes.

*Table 109. Map properties and their corresponding attributes*

| Data Interchange Services client property | Overrides WebSphere Partner Gateway attribute |
|---|---|
| AckReq | Acknowledge Requested |
| Alphanum | Alphanumeric validation table |
| Charset | Char set validation table |

*Table 109. Map properties and their corresponding attributes (continued)*

| Data Interchange Services client property | Overrides WebSphere Partner Gateway attribute |
|---|---|
| CtlNumFlag | Control numbers by Transaction Id |
| EdiDecNot (Decimal notation) | Decimal notation |
| EdiDeDlm (Data element separator) | Data element delimiter |
| EdiDeSep (Repeating data element separator) | Repeating data element separator |
| EdifactGrp | Create Groups for EDI |
| EdiRlsChar (Release character) | Release character |
| EdiSeDlm (Component data element separator) | Subelement delimiter |
| EdiSegDlm (Segment terminator) | Segment delimiter |
| EnvProfName | Envelope profile |
| EnvType | Envelope type |
| MaxDocs | Max Transactions Number |
| Reroute | Interchange reverse routing |
| SegOutput | Segment output |
| ValLevel | Validation level |
| ValErrLevel | Max validation error level |
| ValMap | Validation map |

Table 110 lists additional Data Interchange Services client properties and their associated WebSphere Partner Gateway attributes.

*Table 110. Data Interchange Services client properties and their associated attributes*

| Data Interchange Services client property | Overrides WebSphere Partner Gateway attribute |
|---|---|
| IchgCtlNum | Interchange control number |
| IchgSndrQl | Interchange sender qualifier |
| IchgSndrId | Interchange sender ID |
| IchgRcvrQl | Interchange receiver qualifier |
| IchgRcvrId | Interchange receiver ID |
| IchgDate | Interchange date |
| IchgTime | Interchange time |
| IchgPswd | Interchange password |
| IchgUsgInd | Interchange usage indicator |
| IchgAppRef | Interchange application reference |
| IchgVerRel | Interchange version and release |
| IchgGrpCnt | Number of groups in interchange |
| IchgCtlTotal | Control total from interchange trailer segment |
| IchgTrxCnt | Number of documents in interchange |
| GrpCtlNum | Group control number |
| GrpFuncGrpId | Functional group ID |
| GrpAppSndrId | Group application sender ID |
| GrpAppRcvrId | Group application receiver ID |
| GrpDate | Group date |

*Table 110. Data Interchange Services client properties and their associated attributes (continued)*

| Data Interchange Services client property | Overrides WebSphere Partner Gateway attribute |
|---|---|
| GrpTime | Group time |
| GrpPswd | Group password |
| GrpVer Group version. | Group version |
| GrpRel Group release. | Group release |
| GrpTrxCnt | Number of documents in group |
| TrxCtlNum | Transaction control number |
| TrxCode | Transaction code |
| TrxVer | Transaction version |
| TrxRel | Transaction release |
| TrxSegCnt | Number of EDI Segments in the document |

## AS attributes

This section describes the AS attributes.

*Table 111. AS attributes*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Time to Acknowledge in min | No | The amount of time to wait for an MDN acknowledgment before resending the original request. This attribute works in conjunction with Retry Count. The units are in minutes. | Limited to package or connection | 30 |
| Retry Count | No | The number of times to send a request if an MDN is not received. This attribute works in conjunction with Time to Acknowledge.<br><br>For example, if this attribute is set to 3, the request can potentially be sent four times (the initial time and then the three retries). | Limited to package or connection | 3 |
| AS Compress Before Sign | No | Indicates whether AS compression should be applied to both the payload and signature or only to the payload.<br><br>If you select Yes, the payload is compressed before the message is signed. This attribute works in conjunction with the AS Compressed attribute. | Limited to package or connection | Yes |
| AS Compressed | No | Compress the data. This attribute works in conjunction with the AS Compress Before Sign attribute. | Limited to package or connection | No |
| AS Encrypted | No | This attribute applies to AS2 and is used to specify the URL to which a partner should send an asynchronous MDN. This attribute works in conjunction with the AS MDN Asynchronous attribute, but a value is required even for synchronous MDNs. | Limited to package or connection | No |

*Table 111. AS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| AS MDN Http Url | Yes if the "AS MDN Asynchronous" attribute is Yes and you are using AS2. | This attribute applies to AS2 and is used to specify the URL to which a partner should send an asynchronous MDN. This attribute works in conjunction with the AS MDN Asynchronous attribute, but a value is required even for synchronous MDNs. | Limited to package or connection | |
| AS MDN Email Address | Yes if the "AS MDN Asynchronous" attribute is Yes and you are using AS1. | Specifies the e-mail address for the partner to use when sending an asynchronous MDN. This attribute is used in conjunction with the AS MDN Requested attribute. The value of AS MDN Email Address is used in the “Disposition-notification-to” field.<br><br>For AS1 this attribute works in conjunction with the AS MDN Asynchronous attribute of the format mailto:xxx@company.com.<br><br>For AS2 this attribute still requires a value although the e-mail address itself is not used. | Limited to package or connection | |
| AS MDN Asynchronous | No | Specifies whether the MDN should be returned synchronously or asynchronously. The value of this attribute affects whether the AS MDN HTTP URL or AS MDN Email Address attribute is used.<br><br>Valid values are Yes and No.<br><br>**Yes** Asynchronous<br><br>**No** Synchronous<br><br>If this attribute is Yes, the "receipt-delivery-option" field is filled in based on the AS MDN HTTP URL attribute (for AS2) or the AS MDN Email Address attribute (for AS1). | Limited to package or connection | Yes |
| AS MDN Requested | No | Specifies whether an MDN reply is required. If set to Yes, this attribute causes the “transport Disposition-notification-to” header to be filled in with the value from the AS MDN Email Address attribute.<br><br>Valid values are Yes and No.<br><br>**Yes** Request an MDN.<br><br>**No** Do not request an MDN. | Limited to package or connection | Yes |
| AS Message Digest Algorithm | No | The message digest algorithm to use when signing. This attribute is used in conjunction with the AS Signed and AS MDN Signed attributes.<br><br>For signed MDNs, this value is used to fill in the “Disposition-notification-options: signed-receipt-micalg” header. | Limited to package or connection | sha1 |

*Table 111. AS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| AS MDN Signed | No | Indicates whether the request requires that a signed MDN be returned. This attribute works in conjunction with AS MDN Requested.<br><br>If the value is Yes, the "Disposition-notification-options: signed-receipt-protocol" is filled in.<br><br>Valid values are Yes and No.<br><br>**Yes** Signed MDN requested.<br><br>**No** Signed MDN is not requested.<br><br>If this attribute is set to Yes, the MDN sent by the partner has to be signed.<br><br>If this attribute is set to No, the MDN can be signed or unsigned. | Limited to package or connection | No |
| AS Signed | No | Specifies whether to sign the document.<br><br>For the TO side of an exchange (when you are sending documents to a partner), this specifies whether to sign the document.<br><br>For the FROM side of the exchange (when you are receiving from a partner) if the attribute is set to Yes, an AS request sent from the partner must be signed. If the attribute is set to No, the document from the partner can be signed or unsigned.<br><br>**Yes** Sign the document<br><br>**No** Signed document is not required | Limited to package or connection | No |
| Non-Repudiation Required | No | Indicates whether or not this document needs to be saved in the non-repudiation store. Will apply to the document as both a source or a target.<br><br>Yes – Save the document to the non-repudiation store.<br><br>No – Do not save the document to the non-repudiation store. | Limited to package or connection | Yes |
| Message Store Required | No | Indicates whether or not this document needs to be saved in the message store. Will apply to both the source or target documents.<br><br>Yes – Save the document to the message store.<br><br>No – Do not save the document to the message store. | Limited to package or connection | Yes |

*Table 111. AS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| AS Business Id | No | The AS business ID to use in the “AS2-To” or “AS3-To” header. If a value is not supplied, WebSphere Partner Gateway uses the recipient business ID used in the source document.<br>**Note:** The “AS2-From” or “AS3-From” header will be set from the “AS Business Id” attribute from source document definition or, if not defined, from the original source document that came into WebSphere Partner Gateway and that is being sent out as AS. | Limited to package or connection | |
| AS MDN FTP Address | Yes for AS3 when the "AS MDN Requested" attribute is Yes. | The AS MDN FTP address to use when requesting an MDN. This attribute is used in conjunction with the "AS MDN Requested" attribute. The value of AS MDN FTP Address is used in the “Disposition-notification-to” field. Needs to be in the format: `ftp://username:pwd@host.com:port/folder-name.` | Limited to package or connection | No |
| Signature Algorithm | Yes if "Digital Signature Required" is Yes | The algorithm that is used to sign the document. This attribute is only used if the “Digital Signature Required” attribute value is “Yes.” | | dsa-sha1 |
| Encryption Algorithm | Yes when “Encryption Required” attribute value is set to “Yes” | The algorithm is used to encrypt the payloads. This value works in conjunction with the "Encryption Protocol" attribute.<br><br>This attribute is only used if the “Encryption Required” attribute value is set to “Yes.” | | AES-128 |
| Encryption Protocol | No | The protocol used to encrypt the payloads. The possible values are XMLEncryption and SMIME.<br><br>This attribute is only used if the “Encryption Required” attribute value is set to “Yes”. If EncryptionRequired is set to "yes" and no value is provided for this attribute, then the document . | | XMLEncryption |

## RosettaNet attributes

This section describes RosettaNet attributes.

*Table 112. RosettaNet attributes*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Time To Acknowledge | Yes | The amount of time to wait for a Receipt Acknowledgment before resending the original request. This attribute works in conjunction with Retry Count. The units are in minutes.<br><br>The default value is taken from the RosettaNet PIP specification document. | Limited to package or connection | 120 |

*Table 112. RosettaNet attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Time To Perform | Yes | The amount of time to wait for a response to a request action before sending a failure notification message. | Limited to package or connection | |
| Retry count | Yes | The number of times to send a request when an Receipt Acknowledgment was not received. This attribute works in conjunction with Time to Acknowledge.<br><br>For example, with a setting of 3, the request can potentially be sent 4 times (the initial time and the three retries).<br><br>The default value is taken from the RosettaNet PIP specification document. | Limited to package or connection | 3 |
| Digital Signature Required | No | Indicates whether the PIP message requires a digital signature.<br><br>The default value is taken from the RosettaNet PIP specification document. | Limited to package or connection | Yes |
| Non-Repudiation Required | No | Indicates whether or not this document needs to be saved in the non-repudiation store. Will apply to the document as both a source or a target.<br><br>Yes – Save the document to the non-repudiation store.<br><br>No – Do not save the document to the non-repudiation store. | Limited to package or connection | Yes |
| Message Store Required | No | Indicates whether or not this document needs to be saved in the message store. Will apply to both source or target documents.<br><br>Yes – Save the document to the message store.<br><br>No – Do not save the document to the message store. | Limited to package or connection | Yes |
| Non-Repudiation of Receipt Required | No | Indicates whether to store the Receipt Acknowledgement document in the non-repudiation store.<br><br>The default value is taken from the RosettaNet PIP specification document. | Limited to package or connection | Yes |
| Sync Supported | | Indicates whether the PIP supports synchronous communication.<br><br>The default value is provided based on the PIP specification. | Limited to package or connection.<br><br>This attribute is available for RNIF 2.0 only. | |

*Table 112. RosettaNet attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Sync Ack Required | | Indicates whether the PIP requires a synchronous Receipt Acknowledgment.<br><br>The default value is provided based on the PIP specification. | Limited to package or connection.<br><br>This attribute is available for RNIF 2.0 only. | |
| Global Supply Chain Code | Required for RNIF 1.1 | The code identifying the supply chain for the partner's function.<br><br>Valid values are:<br>• Electronic Components<br>• Information Technology<br>• Semiconductor Technology | Limited to package or connection | |
| Encryption | | This attribute indicates whether encryption should be performed.<br>**Note:** This is not the same as SSL encryption.<br><br>For the TO side of an exchange (when you are sending documents to a partner), this specifies whether to encrypt the document.<br><br>For the FROM side of an exchange (when you are receiving documents from a partner), if the attribute is set to Yes, an RNIF request sent from the partner must be encrypted. If the attribute is set to No, the document from the partner can be encrypted or un-encrypted.<br><br>Valid values are:<br>**None** Encryption is not required.<br>**Payload** Encrypt the RosettaNet Service Content only.<br>**Payload and Container** Encrypt the RosettaNet service content and the service header together. | Limited to package or connection.<br><br>This attribute is available for RNIF 2.0 only. | None |
| Message Standard Text | No | The standard with which the Service content must be compliant. This must be set if and only if this is a non-RosettaNet specified Service Content Message. | | No default value. |
| Message Standard Version | No | The version of the standard with which the service content must be complaint. This must be set if and only if this is a non-RosettaNet specified Service Content Message. | | No default value. |
| PIP Payload Binding Identifier | No | This is partner defined PIP binding identifier, which is unique between the trading partners. This attribute is set only in the case of non-RosettaNet service content. | | No default value. |

*Table 112. RosettaNet attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| FromGlobalPartner ClassificationCode | Yes for RNIF 1.1 schemas | The code identifying a partner's function in the supply chain. Required only when using RNIF 1.1 for Schema based PIPs. This value has to specified for 0A1 pip also, when Schema based PIPs are used. | | No default value. |
| ToGlobalPartner ClassificationCode | Yes for RNIF 1.1 schemas | The code identifying a partner's function in the supply chain. Required only when using RNIF 1.1 for Schema based PIPs. This value has to specified for 0A1 pip also, when Schema based PIPs are used. | | No default value. |
| RN Message Digest Algorithm | No | This attribute is only used when the "Digital Signature Required" attribute is set to Yes. Determines the digest algorithm to use for the digital signature. The values allowed are SHA1 and MD5. | | SHA1 |
| RN Encryption Algorithm | No | This attribute is only used when the "Encryption" attribute is set to "Payload" or "Payload and Container". Allowed values are "Triple DES" and "RC2-40". | | Triple DES |

## Backend Integration attribute

This section describes the attribute associated with Backend Integration packaging.

*Table 113. Backend Integration attribute*

| Attribute | Description | Default |
|---|---|---|
| Envelope Flag | This attribute indicates whether to wrap the document in an XML envelope.<br><br>Valid values are Yes and No. | No |

## ebMS attributes

This section describes the ebMS attributes.

*Table 114. ebMS attributes*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Time to Acknowledge in min | No | The amount of time to wait for an acknowledgment before resending the original request. This attribute works in conjunction with Retry Count. The units are in minutes. | Limited to package or connection | 30 |
| Retry Count | No | The number of times to send a request if an acknowledgment is not received. This attribute works in conjunction with Time to Acknowledge.<br><br>For example, if this attribute is set to 3, the request can potentially be sent four times (the initial time and then the three retries). | Limited to package or connection | 3 |

*Table 114. ebMS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Non-Repudiation Required | No | Indicates whether or not this document needs to be saved in the non-repudiation store. Will apply to the document as both a source or a target.<br><br>Yes – Save the document to the non-repudiation store.<br><br>No – Do not save the document to the non-repudiation store. | Limited to package or connection | Yes |
| Message Store Required | No | Indicates whether or not this document needs to be saved in the message store. Will apply to both source or target documents.<br><br>Yes – Save the document to the message store.<br><br>No – Do not save the document to the message store. | Limited to package or connection | Yes |
| Non-Repudiation of Receipt Required | No | Indicates whether to store the Receipt Acknowledgement document in the non-repudiation store. | Limited to package or connection | Yes |
| Acknowledgment Requested | No | The possible values are always, perMessage, and never.<br><br>If set to “always,” then when sending an ebMS document a request for an acknowledgment will be made by putting an acknowledgmentRequested element in the ebMS SOAP document.<br><br>For sender, “perMessage” and “never” means “No.” When receiving an ebMS document if the value is set as “always”, then the incoming document should request acknowledgment else it will fail.<br><br>If the value is set to “perMessage” on receiver hub, it will not fail the document whether the document requests an acknowledgment or not. If value is set as “never” then the incoming ebMS document should never request an acknowledgment. | | never |

*Table 114. ebMS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Acknowledgment Signature Requested | No | Possible values are always, perMessage and never.<br><br>"always" means request for a signed acknowledgment. "perMessage" and "never" implies that there may be a request for unsigned acknowledgment. This works in conjunction with "AcknowledgementRequested" attribute.<br><br>If the value of AcknowledgmentRequested attribute is set as "perMessage" or "never" then this attribute will not be taken into consideration. .<br><br>If there is no value then "never" will be used. This attribute is only used in sending a document. This attribute is not used for a received document. | | never |
| Actor | No | The attribute is not required to be set in ebMS 2.0 implementation. The actor attribute is needed when a sync acknowledgment is requested. It is put in the ebMS SOAP document.<br><br>The ebMS 2.0 specification suggests a constant value http://schemas.xmlsoap.org/soap/actor/next for this attribute (the default). This is taken care of and user is not required to set this attribute value in any case. It is left to be used in future implementation. | | http://schemas.xmlsoap.org/soap/actor/next |
| Compression Required | No | The possible values are "Yes" and "No." If the ebMS payloads are to be compressed, the value should be set as "Yes." If compression is not required, do not set anything or set is as "No." | | No |

*Table 114. ebMS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Duplicate Elimination | No | While sending an ebMS message, if this attribute value is set to “always”, then it will place a DuplicateElimination element in the ebMS SOAP document. This DuplicateElimination element in ebMS SOAP document indicates that the receiver hub will not deliver the ebMS payloads to backend if the ebMS document is a duplicate.<br>**Note:** For a SOAP document, values “perMessage” and “never” will not be placed in the DuplicateElimination element.<br><br>While receiving an ebMS document, if the value is set to “always”, then DuplicateElimination element has to be in the ebMS SOAP document, else it will fail the document. If the value set is "perMessage" and if the received document has duplicateElimination element set, then the duplicate check has to be done.<br><br>For a received ebMS document, if the attribute value is “always, and if the DuplicateElimination element is present, the document will be checked to see if it is a duplicate. If the document is a duplicate, then the document will be failed.<br><br>For value "never", if the DuplicateElimination element is present in the SOAP document, then the document will fail.<br><br>If there is no value, then “never” will be used. | | never |
| Encryption Constituent | No | The value of this attribute should be a list of semicolon separated content type for payloads, for example application/xml;text/xml; application/binary:application/edi will cause payloads with these content types will be encrypted.<br><br>This attribute is only used if the “Encryption Required” attribute value is set to “Yes.” | | application/ xml;text/xml; application/ EDI-X12; application/ EDI-CONSENT; application/ EDIFACT; application/ binary; application/ octet-stream |
| Encryption Mime Parameter | No | An optional attribute used for putting additional parameters as MimeMultipart headers to the encrypted document. Will apply to each encrypted payload. Example value: smime-type=”enveloped-data” or type=”text/xml” version=”1.0.”<br><br>This attribute is only used if the “Encryption Required” attribute value is set to “Yes.” | | No default value.<br><br>**Note:** This variable is not used in current implementation. Setting this will not take effect in the runtime. |
| Encryption Mime Type | No | Not used in current implementation. | | No default value |

*Table 114. ebMS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Encryption Required | No | Possible values are “Yes” and “No.” If set to "Yes," the payloads will be encrypted. This attribute works in conjunction with “Encryption Constituent.”<br>**Note:** If “Encryption Required” is set to “Yes” and there are not content types configured for “Encryption Constituent,” nothing will be encrypted. | | |
| Encryption Transformation | No | Not used in current implementation. | | No default value.<br><br>**Note:** This variable is not used in current implementation. Setting this will not take effect in the runtime. |
| Exclude from Signature | No | The value of this attribute will be a list of semicolon separated content types, for example: `application/binary;application/octet-stream`. Payloads having this content type will not be included in the signature.<br><br>This attribute is only used if the “Digital Signature Required” attribute value is “Yes.” | | No entries, will apply signature to all payloads. |
| Hash Function | No | Hash algorithm that should be used in the XML Signature when hashing the payloads during signing. This attribute is only used if the “Digital Signature Required” attribute value is “Yes.”<br><br>Only SHA1 is supported as Hash Algorithm for ebMS. Even if any other hash algorithm is set in the connection for ebMS documents, SHA1 is used as the hash algorithm. | | SHA1 |
| Message Order Semantics | No | The possible values are “Guaranteed” and “NotGuaranteed”. When sending a document, if the value is set to “Guaranteed” then a Message Order element will be put in the SOAP document. The receiving hub on identifying this element in the SOAP document will make sure that the payloads are delivered to backend in sequence.<br><br>For a received document if this attribute is set to “Guaranteed” then the incoming ebMS document should have MessageOrder element present in it and if missing the document will be failed and an error message with errorCode “Inconsistent” will be sent to the partner. | | NotGuaranteed |

*Table 114. ebMS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Role | No | When sending an ebMS document this attribute value is as role element value in the ebMS SOAP document.<br><br>When receiving an ebMS this attribute value is compared with the role element value in ebMS SOAP document and if the values do not match (even if the attribute value is empty) then the document is failed and an error message with errorCode “Inconsistent” is sent to the partner. | | No default value |
| Persist Duration | No | The time in minutes for which the document should be persisted, for example 1440 for 24 hours.<br><br>When sending a document, the Persist Duration is used to calculate TimeToLive using the formula: TimeToLive = Persist Duration + (no. of Retries * RetryInterval).<br><br>When receiving a document, the Persist Duration is used to do duplicate elimination. If a document with the duplicate messageID is received, it is checked whether or not the Persist Duration for the earlier document is passed. If the persist duration is not passed, document is marked as duplicate else the document is not marked as a duplicate.<br><br>If no entry then the value defaults to 0. | | 0 |
| Packaging Constituent | No | Not used in current implementation. | | No default value.<br><br>**Note:** This variable is not used in current implementation. Setting this will not take effect in the runtime. |
| Package Mime Parameter | No | Not used in current implementation. | | No default value.<br><br>**Note:** This variable is not used in current implementation. Setting this will not take effect in the runtime. |
| Encryption Algorithm | Yes when “Encryption Required” attribute value is set to “Yes” | The algorithm is used to encrypt the payloads. This value works in conjunction with the "Encryption Protocol" attribute.<br><br>This attribute is only used if the “Encryption Required” attribute value is set to “Yes.” | | AES-128 |

*Table 114. ebMS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Encryption Protocol | No | The protocol used to encrypt the payloads. The possible values are XMLEncryption and SMIME.<br><br>This attribute is only used if the “Encryption Required” attribute value is set to “Yes”. If EncryptionRequired is set to "yes" and no value is provided for this attribute, then the document will fail. | | XMLEncryption |
| Retry Interval | No | For a sent document the time interval in minutes to for an acknowledgment before resending the ebMS document. ebMS Documents are resent only when an acknowledgment is requested but an acknowledgment has not been received from the partner within the retry interval.<br><br>A value of 0 indicates that there will not be any retries. This attribute works in conjunction with the “Retry Count” attribute. | | 270 |
| Signature Algorithm | Yes if "Digital Signature Required" is Yes | The algorithm used to sign the document. This attribute is only used if the “Digital Signature Required” attribute value is “Yes.”<br>**Note:** In ebMS, hmac-sha1 is not supported. | | dsa-sha1 |
| Signature Transformation | No | The transformation algorithm used to transform the payloads before creating the XML Signature. This attribute is only used if the “Digital Signature Required” attribute value is “Yes.” | | No default value |
| Sync Reply Mode | No | The type of synchronous response required for the document that is being sent.<br><br>If the value is set as:<br>• **MSHSignalsOnly** - Only MSH acknowledgment/error documents will be sent over a synchronous connection. The business response and business signal documents will be returned asynchronously.<br>• **signalsOnly** – Only Business signal documents and MSH documents will be sent over a synchronous connection. The business response will be returned asynchronously.<br>• **responseOnly** – Only business responses and MSH documents will be sent over a synchronous connection. Business signal documents will not be returned.<br>• **signalsAndResponse** - Business responses and business signals documents will be sent over a synchronous connection.<br>• **none** – No synchronous response documents from the receiver. | | none |

*Table 114. ebMS attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Intelligible Check Required | No | The value of this attribute is sent to backend as the value of header "x-aux-IntelligibleCheckRequired". The possible values are "yes" and "no." Purpose is to indicate to the backend that the ReceiptAcknowledgement should only be sent if the ebXML document with the payloads contains no error. It is up to backend to interpret this value. | | No |
| Canonicalization Method | No | The Canonicalization algorithm used before doing XML Signature. This attribute is only used if the "Digital Signature Required" attribute value is "Yes." | | INCLUSIVE_ WITH_ COMMENTS |
| Compression Constituent | No | The list of semicolon-separated content type of payloads, which are to be compressed. For example if payloads with contentType "text/xml" and "application/edi" are needed to be compressed then the value of this attribute will be "text/xml;application/edi". No entries means no payloads will be compressed even with "Compression Required" is set to "Yes."<br><br>This attribute is only used if the "Compression Required" attribute value is "Yes." | | application/xml; text/ xml;application/ EDI-X12; application/ EDI-CONSENT; application/ EDIFACT |
| Service Type | Yes if the Service element (Document Type) value is not a URI | When sending an ebMS document, the ebMSService element value in ebMS SOAP message has to be either a URI or some string. In case it is a string, this type attribute is required. If the Service (Document Type) value is not a URI then this Service Type attributes value is used as type attribute value in ebMS document. | | No default value |

## General attributes

This section describes the general attributes.

*Table 115. General attributes*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| Validation Map | No | The validation map to use for validating this document. The Action that is used at runtime has to have a validation step that makes use of this attribute. Only Validation maps that have been uploaded and associated with this Document Type will be selectable. | Limited to package or connection | No default value. |
| User Attribute 1 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User01 as either a From (Source document) or To (Target Document) prefix. | | No default value. |

*Table 115. General attributes (continued)*

| Attribute | Required | Description | Restrictions | Default |
|---|---|---|---|---|
| User Attribute 2 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User02 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 3 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User03 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 4 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User04 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 5 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User05 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 6 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User06 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 7 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User07 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 8 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User08 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 9 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User09 as either a From (Source document) or To (Target Document) prefix. | | No default value. |
| User Attribute 10 | No | For use in User defined exits. The value is determined by the creator of the User defined exit. These will be set in the Business Document Object with attribute bcg.ro.user.User10 as either a From (Source document) or To (Target Document) prefix. | | No default value. |

## OpenPGP attributes

After creating a connection between external and internal partner, set the connection attributes as described in this topic.

In the **Manage Connections** page, after you activate a connection, click **Attributes** at the target side of the B2B Capabilities to set values for OpenPGP specific connection attributes.

The various OpenPGP connection attributes are as follows:

*Table 116. OpenPGP attributes*

| Attribute | Required | Description | Default |
|---|---|---|---|
| Use OpenPGP format | Yes | Set the target side of the B2B Capabilities in the Manage Connections page to "Yes" to use the OpenPGP format. | No default value. |
| Encryption required | Optional | You can use this attribute for encrypting the payloads. For encrypting the payloads, set the value to "Yes". | No default value. |
| Symmetric algorithm preference | Mandatory | This attribute is the preferred key algorithm to be used for OpenPGP encryption. From the drop-down list, select the preferred Symmetric Algorithm preference. When the attribute **Encryption Required** is set to true, it is mandatory to set this attribute. | No default value. |
| Modification detection | Optional, and is selected only with encryption | If you want to enforce message integrity check, set this attribute to true. This setting verifies whether the message has been tampered during transition or not. | No default value. |
| Compression required | Optional | You can use this attribute to compress the payloads. Set this attribute to Yes for compression. | No default value. |
| Compression algorithm preference | Mandatory | This attribute is the preferred compression algorithm for OpenPGP. From the drop-down list, select the preferred compression algorithm. When the attribute **Compression Required** is set to true, it is mandatory to set this attribute. | No default value. |
| Armor | Optional | OpenPGP encodes data into ASCII Armor. It places specific headers around the Radix-64 encoded data, so that OpenPGP can reconstruct the data later on. ASCII armor is also used to protect raw binary data when transferred over the wire. If you set it to true at the target side of the connection, armor is carried out in the document packaging. Setting a value for this attribute is optional. | No default value. |

# 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, N.Y. 10504-1785*
*U.S.A.*

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*IBM World Trade Asia Corporation Licensing*
*2-31 Roppongi 3-chome, Minato-ku*
*Tokyo 106-0032, 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 want 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 Burlingame Laboratory Director
IBM Burlingame Laboratory
577 Airport Blvd., Suite 800
Burlingame, CA 94010
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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:



If you are viewing this information softcopy, the photographs and color illustrations may not appear.

## Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program. General-use programming interfaces allow you to write application software that obtain the services of this program's tools. However, this information also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

**Attention:** Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

## Trademarks and service marks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

| | | | | |
|---|---|---|---|---|
| IBM | DB2 | IMS | MQIntegrator | Tivoli |
| the IBM logo | DB2 Universal Database | Informix | MVS | WebSphere |
| AIX | Domino | iSeries | OS/400 | z/OS |
| CICS | IBMLink | Lotus | Passport Advantage | |
| CrossWorlds | i5/OS | Lotus Notes | SupportPac | |

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

MMX, Pentium, and ProShare are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

Solaris, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

WebSphere Partner Gateway Enterprise and Advanced Editions includes software developed by the Eclipse Project (www.eclipse.org)

# Index

## Special characters



## Numerics



## A

# B



# C



# D

# E

# F

## G



## H



## I



## J



## K



## L



## M

## N



## O



## P

## Q



## R



## S

# T



# U



# V



# W

# X



# Z

Printed in USA