IBM WebSphere Business Integration Connect Enterprise
and
Advanced Editions

Hub Configuration Guide

Version 4.2.2

IBM WebSphere Business Integration Connect Enterprise
and
Advanced Editions

Hub Configuration Guide

Version 4.2.2

Notices:

Before using this information and the product it supports, be sure to read the general information under “Notices and Trademarks” on page 145.

29June 2004

This edition of this document applies to IBM WebSphere Business Integration Connect Enterprise Edition (5724-E87) and Advanced Edition (5724-E75), version 4.2.2.

To send us your comments about this document, e-mail doc-comments@us.ibm.com. We look forward to hearing from you.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

New in this release

New in release 4.2.2

Version 4.2.2 is the first release of the *Hub Configuration Guide* .

Preface

About this book

This document describes how to configure the IBM (R) WebSphere (R) Business Integration Connect server.

Audience

This document is intended for the person responsible for configuring the WebSphere Business Integration Connect server, also known as the hub. To configure the hub, you should be the Hub Admin. The Hub Admin has the ability to use all the features of the WebSphere Business Integration Connect Community Console to configure and operate the hub.

Typographic conventions

This document uses the following conventions.

`courier font`	Indicates a literal value, such as a command name, filename, information that you type, or information that the system prints on the screen.
bold	Indicates a new term the first time that it appears.
italic, *`italic`*	Indicates a variable name or a cross-reference.
blue outline	A blue outline, which is visible only when you view the manual online, indicates a cross-reference hyperlink. Click inside the outline to jump to the object of the reference.
{ }	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 an optional parameter.
...	In a syntax line, ellipses indicate a repetition of the previous parameter. For example, `option[,...]` means that you can enter multiple, comma-separated options.
< >	In a naming convention, angle brackets surround individual elements of a name to distinguish them from each other, as in `<server_name><`*`connector_name`*`>tmp.log`.
/, \	In this document, backslashes (\) are used as the convention for directory paths. For UNIX installations, substitute slashes (/) for backslashes. All IBM WebSphere InterChange Server product pathnames are relative to the directory where the IBM WebSphere InterChange Server product is installed on your system.
`%text%` and *`$text`*	Text within percent (%) signs indicates the value of the Windows `text` system variable or user variable. The equivalent notation in a UNIX environment is *`$text`*, indicating the value of the *`text`* UNIX environment variable.
ProductDir	Represents the directory where the product is installed.

Related documents

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

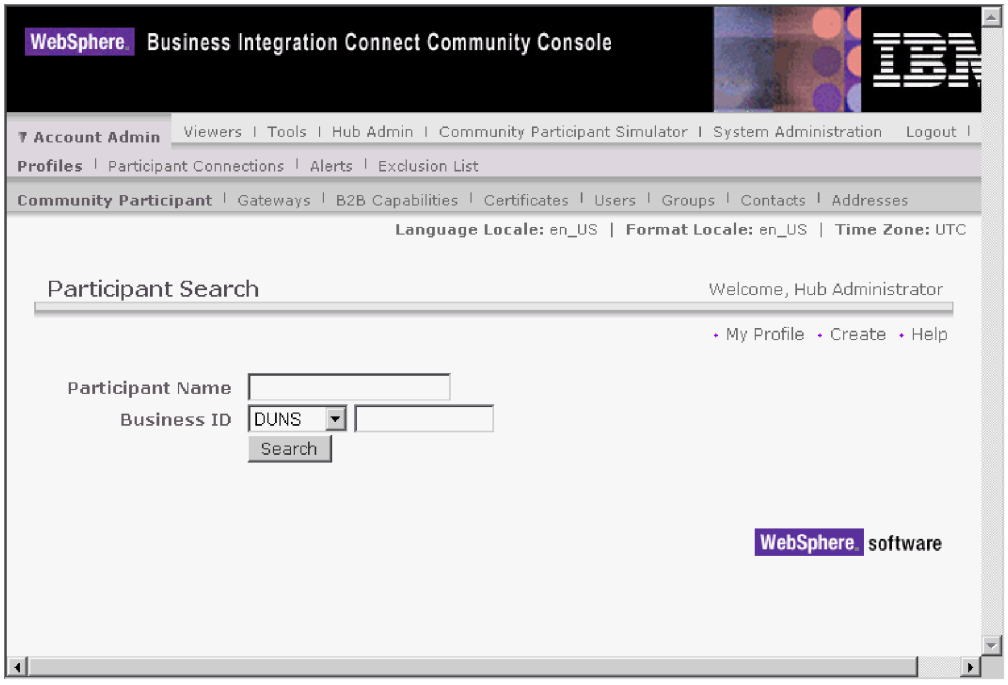
You can download, install, and view the documentation at the following site:
http://www.ibm.com/software/integration/wbiconnect/library/infocenter

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/wbiconnect/support

Getting help

Online help

Click the **Help** link to access the online help.



Software support

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

Passport advantage

http://www.ibm.com/software/howtobuy/passportadvantage/

Product documentation

http://www.ibm.com/software/integration/wbiconnect/library/infocenter

Chapter 1. Introduction

After you install WebSphere Business Integration Connect and before any documents can be exchanged between the Community Manager and participants, you must configure the WebSphere Business Integration Connect server (the hub).

The goal is to enable the Community Manager to send a document (electronically) to a participant or to receive a document from a participant. 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.

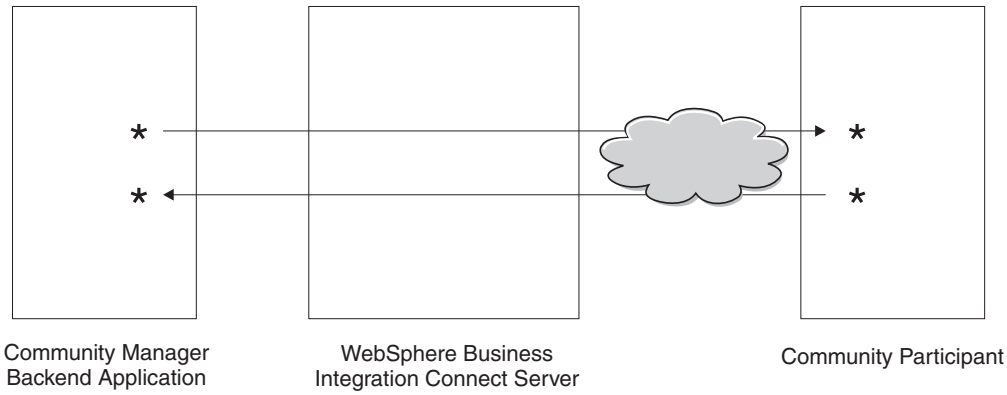


Figure 1. How documents flow through the hub

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

Information needed to set up the hub

You need some information about the types of exchanges in which the Community Manager 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 Community Manager and its participants be sending through the hub?
- Which types of transports (for example, HTTP or FTP) will the Community Manager and its participants use to send the documents?
- Will the documents be transformed before being delivered?
- Will the documents be validated before being delivered?
- Will the documents be encrypted or digitally signed or use some other security technique?

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

After you define the hub, you can define your participants, using information (such as IP address and DUNS numbers) that is supplied to you by the participants.

An overview of document processing

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

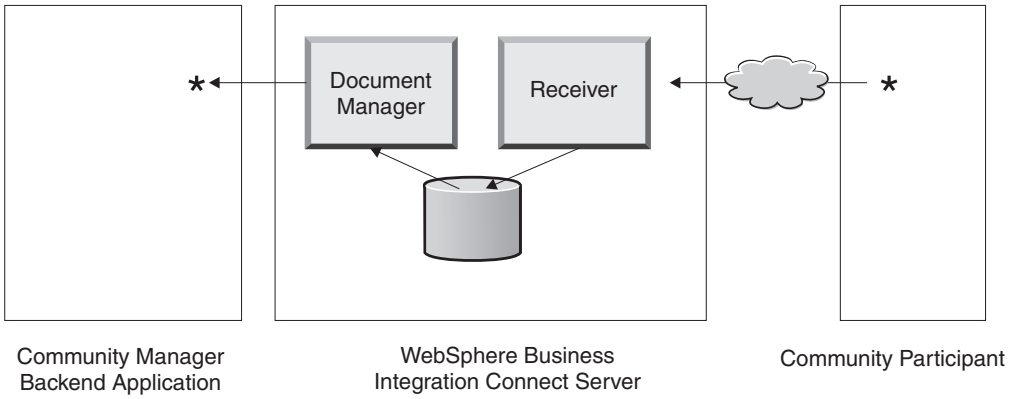


Figure 2. The Receiver and Document Manager components

This illustration is an example of how a document is sent from a participant, received at the hub, processed at the hub, and sent to a Community Manager backend application.

A document is received into the WebSphere Business Integration Connect server by the Receiver component. The Receiver includes transport-specific targets. You set up a target for each type of transport the hub will support. For example, if participants are going to send documents over HTTP, you set up an HTTP target to receive them. As you will see in the section on Gateways, you set up a gateway for the transport type used to send the document from the hub to the Community Manager.

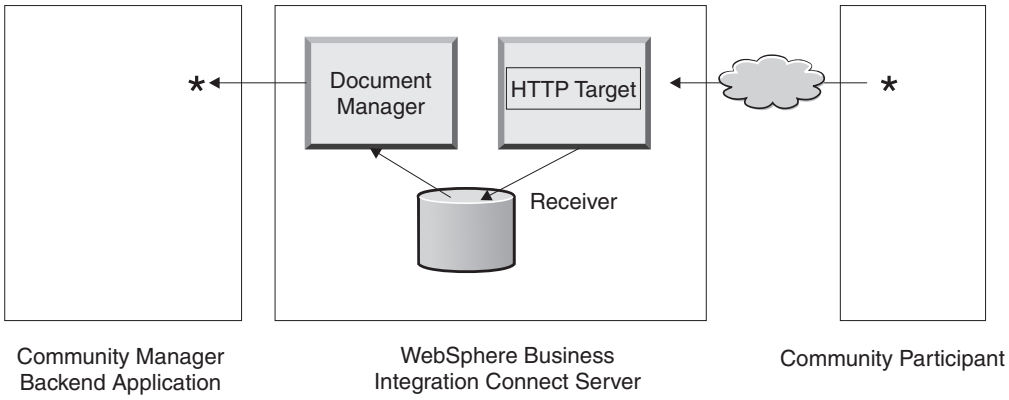


Figure 3. An HTTP target

If the Community Manager backend application is going to send documents over JMS, you set up a JMS target at the hub to receive them. You will also set up a gateway for the transport type used to send the document from the hub to the participant.

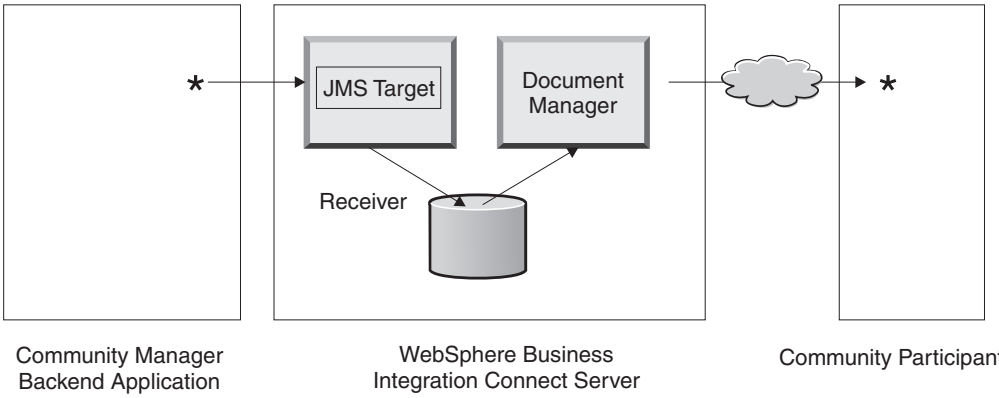


Figure 4. A JMS target

WebSphere Business Integration Connect supports a variety of transports, but you can also upload your own user-defined transport and use it when defining a target (as described in Chapter 5).

The Receiver sends the document to a 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 Community Manager might send an EDI-X12 document with no packaging to the hub, for delivery to a participant that is expecting the EDI-X12 document to include AS2 headers. The Document Manager adds the header information and then uses the gateway defined for the participant to send the document to its destination.

Configuring document processing components with handlers

This section describes, in more detail, the components of WebSphere Business Integration Connect and shows you the various points at which you can change the system-supplied behavior of the components for processing a business document.

You use *handlers* to change the system-supplied behavior of targets, gateways, fixed workflow steps, and actions. There are two types of handlers--those supplied by WebSphere Business Integration Connect and those that are user-defined. See the *Programmer Guide* if you want information on creating handlers.

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

Targets

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

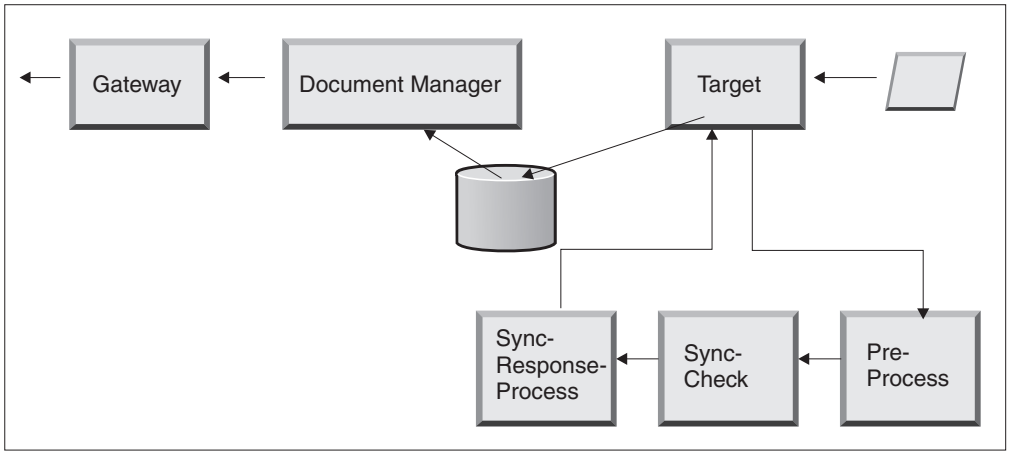


Figure 5. Target configuration points

Preprocessing is generally used for any processing on the document (for example, splitting the document) that needs to be accomplished before the document is sent to the shared file system.

SyncCheck is used to determine whether the document is synchronous or asynchronous. WebSphere Business Integration Connect supplies the following handlers for synchronous checking:

- 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

As you can see from the naming convention, the first four handlers are specific to the four transports that can be used for synchronous transactions. Any request that uses the DefaultAsynchronousSyncCheckHandler will be treated as an asychronous request. Any request that uses the DefaultSynchronousSyncCheckHandler will be treated as a sychronous request.

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

For the HTTP/S transport and for user-defined transports, you can add handlers to be called at the three configuration points available for targets. In the case of AS2, cXML, RNIF, and SOAP documents, you must specify the SyncCheck handler. This is described in “Modifying configuration points” on page 33.

When you select a configuration point during the creation of an HTTP/S or user-defined target, you see two lists of handlers: a Configured List and an Available List. The Configured List shows any handlers that have been configured for the target. The Available List shows any handlers that can be used to configure the target.

You manipulate the handlers in the Configured List by highlighting a handler and using the control buttons (such as **Move Up** or **Move Down**).

The following illustration shows the list of available handlers for the SyncCheck configuration point.

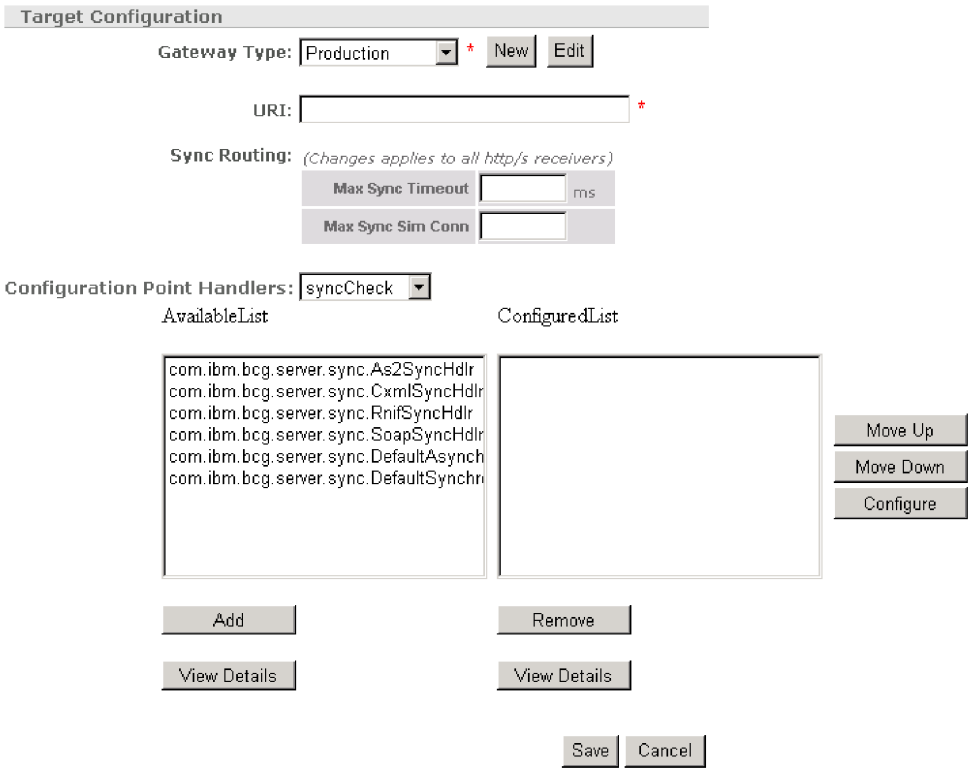


Figure 6. The Available and Configured Lists

You can add your own handler to the handlers supplied by the system by uploading a user-defined target handler. You use the **Import** choice of the Handlers List page to upload a user-dined handler.

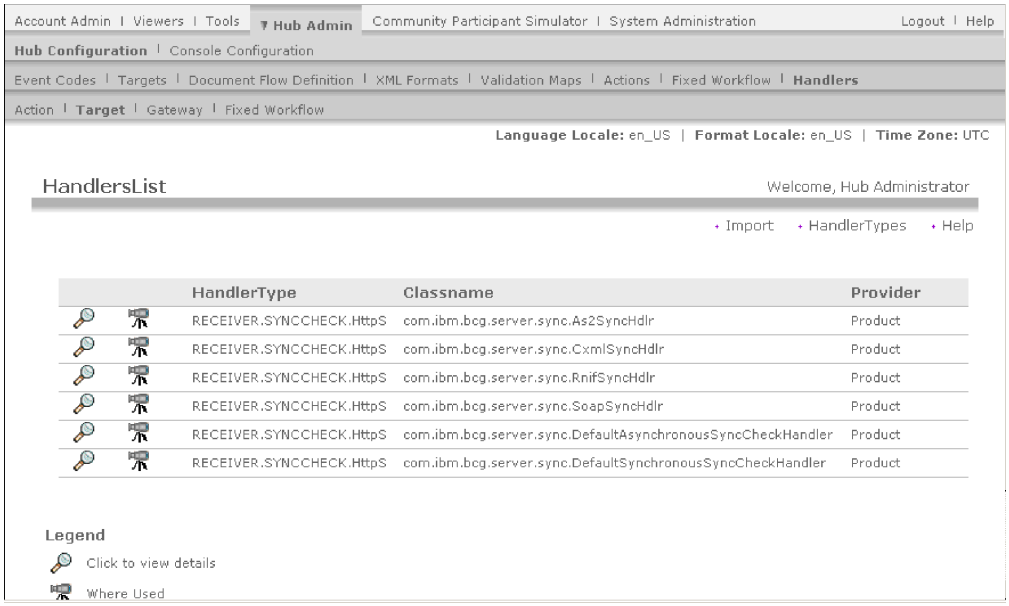


Figure 7. The Handlers List

When you upload a user-defined target 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.

You can move handlers from the Available List to the Configured List, you can remove handlers from the Configured List, or you can rearrange the order of the handlers.

Note: Handlers are invoked in the order in which they appear on the Configured List, but the first handler is not always the one used to configure the target. It is the first *available* handler (the first handler able to process the request) that is used. For example, suppose a target has three handlers configured (Handler1, Handler2, and Handler3, in that order). If a request is made for a handler, the first handler that responds to the request is the one that processes it, and any handlers after it (in the Configured List) are not called. In the example, if Handler2 responds first, Handler3 is never called.

Document Manager

When a document is sent, by the Target, to the shared file system, the Document Manager is triggered to pick up that document for processing. All document processing, regardless of the packaging, protocol, and document flow, involves the use of fixed inbound workflow steps, one or more actions (variable workflow steps), and a fixed outbound workflow step.

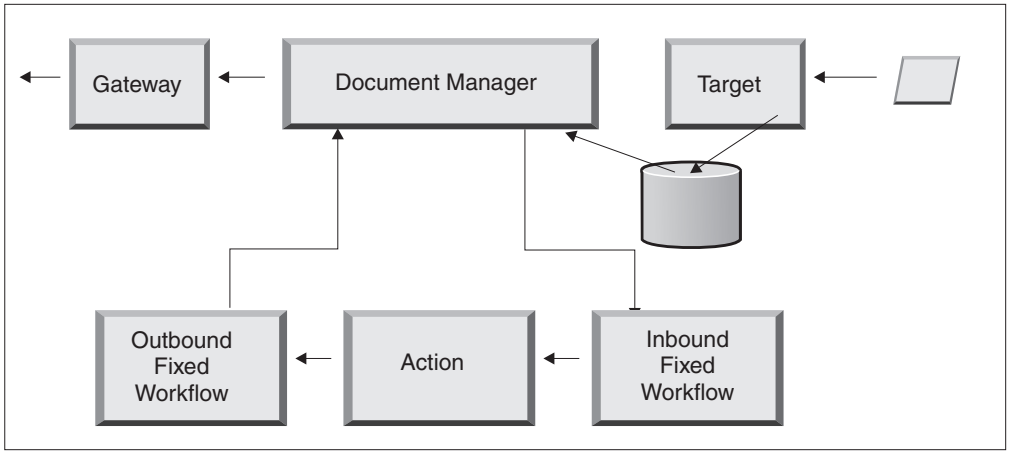


Figure 8. Fixed workflows and actions

Inbound fixed workflow

The Inbound Fixed Workflow consists of two steps that unpackage the protocol and parse the 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 Business Integration Connect and extract information so that the action for the message can be determined.

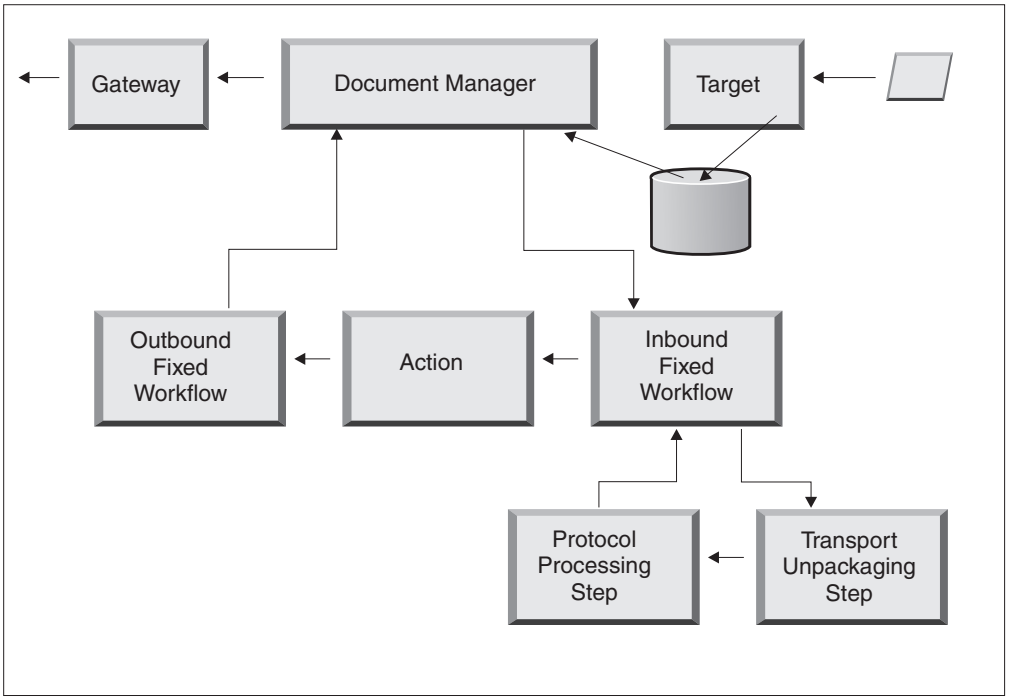


Figure 9. Inbound fixed workflow steps

The business protocol of the document determines how the two steps retrieve this information. At minimum, the document or message must include the sender and receiver IDs and the document flow definition (package, protocol, and document flow).

You can use the default handler that applies to the protocol for your document, or you can specify a different handler for the fixed workflow step.

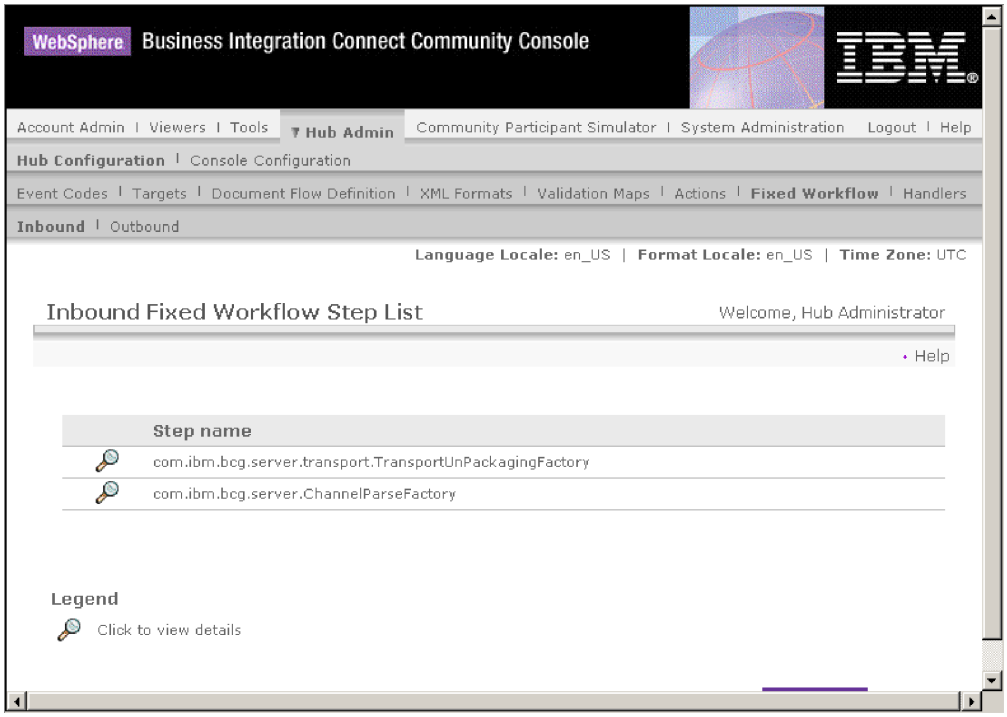


Figure 10. The Inbound Fixed Workflow Step List

After you click the magnifying glass icon, you see the handlers that you can select for each of the inbound fixed workflow steps:

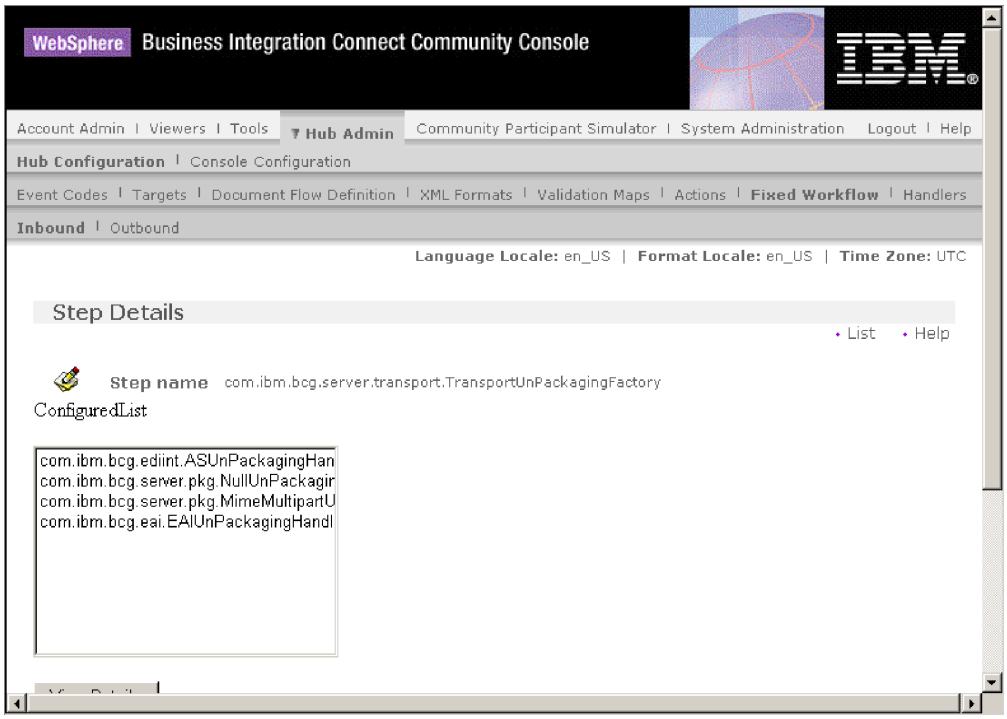


Figure 11. The Step Details page

The fixed workflow steps that are preconfigured with the system are shown in the Configured List. You cannot modify these steps; however, you can add business logic to the steps by adding handlers.

To add user-defined handlers for a fixed inbound workflow step, you upload the file representing the handler. After the file has been uploaded, it appears on the Available List of handlers, and you can add it to the Configured List.

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 system-supplied Pass through action, which makes no changes to the document.

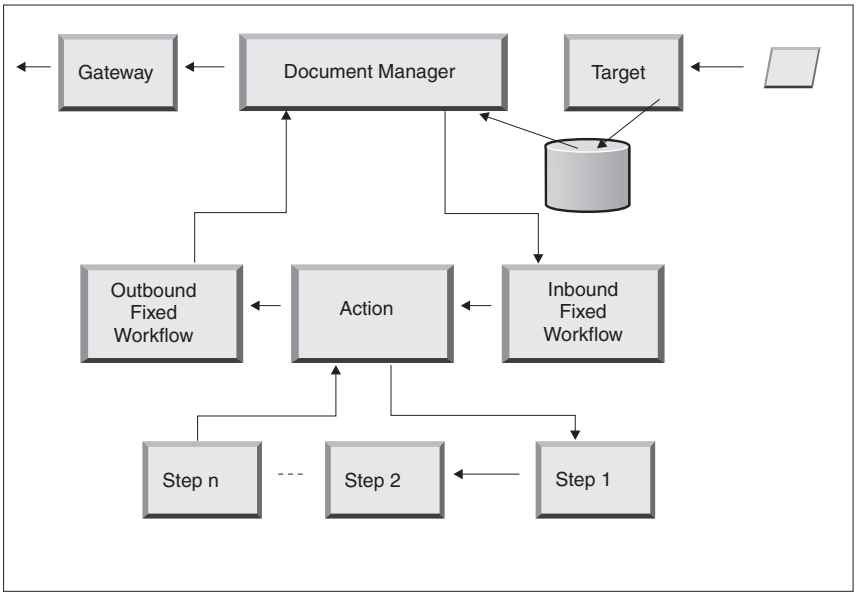


Figure 12. Action steps

The way handlers are processed for actions is different from the way they are processed for targets, gateways, and fixed workflows. For actions, *all* handlers in the Configured List are called, and all are used in the order in which they appear on the list.

Outbound fixed workflow

The Outbound Fixed Workflow consists of one step—the packaging of the document with its protocol information. For example, if this 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 gateway.

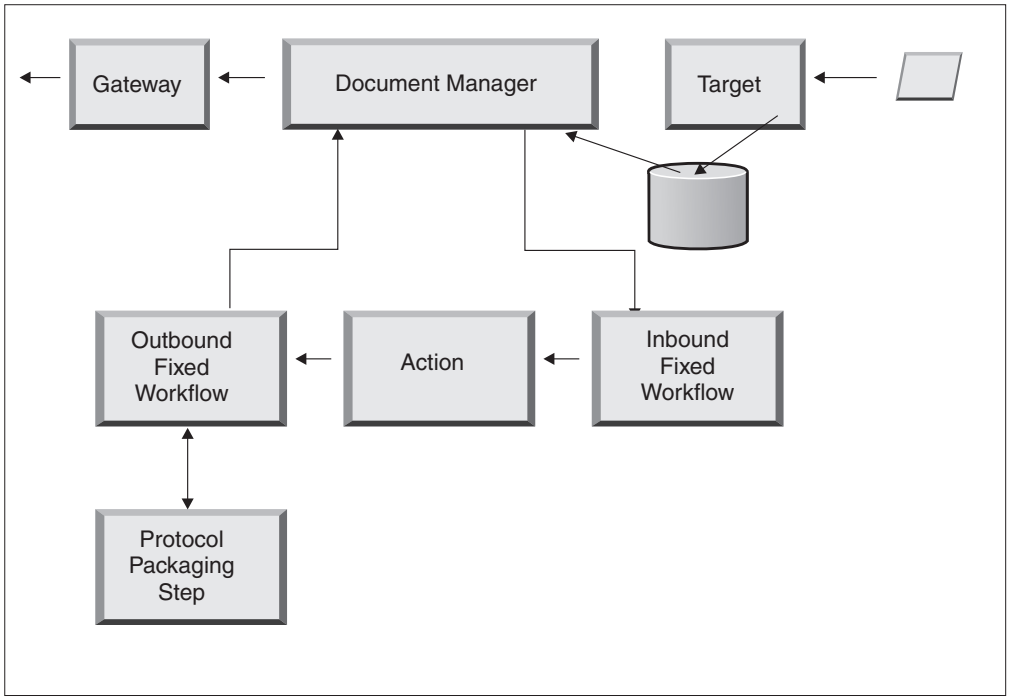


Figure 13. Outbound fixed workflow steps

You can view the system-supplied outbound workflow steps by selecting **Hub Configuration > Fixed Workflow > Outbound**. To upload a user-defined handler to add to the list of system-supplied handlers, you select **Hub Configuration > Handlers > Fixed Workflow** and then select **Import** to upload the user-defined handler.

Gateways

After the document leaves the Document Manager, it is sent, from the Gateway, to the intended recipient. The Gateway has two configuration points—pre-process and post-process.

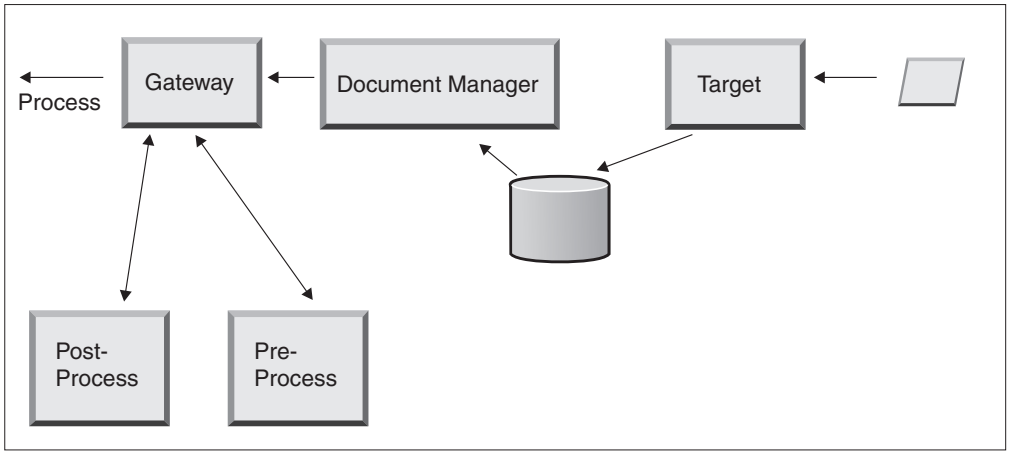


Figure 14. Gateway configuration points

Preprocess affects the processing of a document before it is sent to the recipient. Process is the actual sending of the document. Postprocess acts on the results of

the document transmission (for example, on the response it receives from the recipient during a synchronous transmission).

There are no requirements to set up configuration handlers for any WebSphere Business Integration Connect-supported protocols when you define a gateway (as there are in the case of certain business protocols used in synchronous transactions when you set up targets).

As you set up targets, gateways, and document flows in the next few chapters, you will see how you can (or must) specify a handler for a specific configuration point. If you are going to apply user-defined handlers to the configuration points, you must first upload the files representing those handlers into the hub. This is described in "Uploading user-defined handlers" on page 29.

Note: Handlers supplied by WebSphere Business Integration Connect do not have to be uploaded.

Chapter 2. Preparing to configure the hub

In the next few chapters, you will be setting up the targets and gateways described in Chapter 1, “Introduction”. Depending on the types of transport you will be using to receive documents into targets and to send them from gateways, you have to do some setup work.

This chapter is intended for anyone who is going to be setting up the following types of gateways or targets:

- A file-directory gateway
- A JMS target
- An FTP target

If you are not planning to set up any of these types of targets or gateways, skip this chapter and go to Chapter 3, “Starting the server and displaying the Community Console.”

Creating a directory for a file-directory gateway

If you are going to use a file-directory gateway to send documents to the Community Manager, you must first create a directory on the file system used by the Community Manager.

For example, suppose you wanted to create a directory named FileSystemGateway under the c:\temp directory of a Windows installation. These are the steps you would perform:

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

Configuring the FTP server for receiving documents

Note: This section applies only to receiving documents over FTP or FTPS from participants. Sending documents to participants is described in “Creating an FTP gateway” on page 52 and “Creating an FTPS gateway” on page 55.

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 Business Integration Connect is installed.
- The bcguser on the WebSphere Business Integration Connect machine has read/write access to the location where the FTP server will be storing files.

Configuring the required directory structure on the FTP server

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 Business Integration Connect requires a particular directory structure so that the Receiver

and Document Manager components can correctly identify the participant sending the incoming document. The structure looks like this:

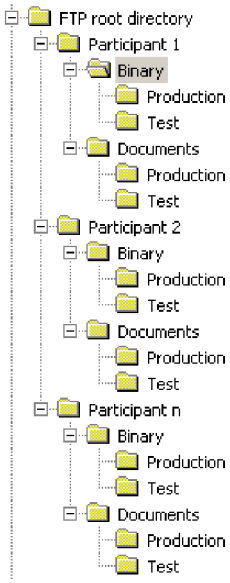


Figure 15. FTP Directory structure

Each participant 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 participant sends an XML document containing complete routing information (using FTP) to the hub. This requires the creation of a Custom XML definition. See "Managing custom XML" on page 41.

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

For each participant 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 participant.
2. Create subfolders under the participant folder named **Binary** and **Documents**.
3. Create subfolders under the Binary and Documents folders called **Production** and **Test**.

How files sent over FTP are processed

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: *<ToParticipantID><UniqueFileName>*

When a binary file is detected by the Receiver, 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 Participant Name, and the first part of the file name is used to evaluate the To Participant 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 Participant Name** is **partnerTwo** (because the file was found in the partnerTwo part of the directory tree).
- The **To Participant Name** is **partnerOne** (because the first part of the file name is 123456789, which is the DUNS ID for partnerOne).
- The Transaction type is Production.

The Document Manager then looks for a Production participant connection from partnerTwo to partnerOne for **None (N/A)/Binary (1.0)/Binary (1.0)** and processes the file.

XML files

An XML file 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 an XML file is detected by the Receiver, 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. The From Participant Name, To Participant 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 participant connection before processing the file.

Note: Files such as EDI documents, when received over FTP, are processed as Binary by the Document Manager. These documents are treated by the WebSphere Business Integration Connect system as pass-through documents.

Additional FTP server configuration

After creating the required directory structure, you configure your FTP Server for each of the participants 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:

1. Add a new group (for example, WBIC).
2. Add a user to the newly created group for each participant who will be sending or receiving documents over FTP.
3. For each participant, set up the FTP server to map the incoming participant to the respective directory structure you created for the participant in the earlier section “Configuring the required directory structure on the FTP server” on page 13. 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 participant is using. There is no specific security configuration for WebSphere Business Integration Connect on incoming FTPS documents. WebSphere Business Integration Connect retrieves the documents from the FTP target (which is described in "Setting up an FTP target" on page 32) 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 participant can contact.

Configuring the hub for the JMS transport protocol

You installed WebSphere MQ as part of the installation of WebSphere Business Integration Connect. WebSphere MQ includes a JMS implementation, which you can use to set up JMS communication.

WebSphere MQ is not configured for JMS by default, however. This section provides the steps to configure JMS.

Creating a directory for JMS

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:

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

Modifying the default JMS configuration

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." 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

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

```
set PATH=%PATH%;<path to installation directory>\_jvm\jre\bin
```

where *installation directory* refers to the directory where WebSphere Business Integration Connect is installed.

Defining the JMS configuration

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(jms)
   change ctx(jms)
   ```
4. From the InitCtx/jms> prompt, enter the following JMS configuration:

```
define qcf(WBICHub)
    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
```

As an example, the following is the JMSAdmin session used to define the queue connection factory as WBICHub, 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 16. Note that user input follows the > prompt.

```
InitCtx> define ctx(jms)
InitCtx> change ctx(jms)
InitCtx/jms> define qcf(WBICHub)
    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
```

Chapter 3. Starting the server and displaying the Community Console

This chapter shows you how to start the WebSphere Business Integration server and display the Community Console.

Starting WebSphere MQ

If you have not already done so, start WebSphere MQ by following one of these procedures:

- For Unix-based systems:
  1. Enter:
     ```
     su mqm
     ```
  2. Enter:
     ```
     strmqm bcg.queue.manager
     ```
  3. Enter:
     ```
     runmglsr -t tcp -p 9999 -m bcg.queue.manager &
     ```
  4. Wait about 10 seconds and press Enter to return to the command prompt.
  5. Enter:
     ```
     strmqbrk -m bcg.queue.manager
     ```
- For Windows-based systems:
  1. Enter:
     ```
     strmqm bcg.queue.manager
     ```
  2. Enter:
     ```
     runmglsr -t tcp -p 9999 -m bcg.queue.manager
     ```
     The Listener runs in this window, so leave it open.
  3. Open a new window and start the JMS Broker (the publish-subscribe broker) with the following command:
     ```
     strmqbrk -m -bcg.queue.manager
     ```

Starting the WebSphere Business Integration Connect components

To start the server, you must start each of the three components of WebSphere Business Integration Connect: the Console, the Document Manager, and the Receiver.

1. Change to the directory \IBM\WBIConnect\console\was\bin.
2. Type the following command to start the Console:
   - For Unix-based systems:
     ```
     /startserver server1
     ```
   - For Windows-based systems:
     ```
     startserver server1
     ```
3. When you see the message:
   ```
   Server server1 open for business
   ```
   change to the directory IBM\WBIConnect\receiver\was\bin
4. Type the following command to start the Receiver:

```
startserver server1
```

or

```
/startserver server1
```

5. When you see the message:

   ```
   Server server1 open for business
   ```

   change to the directory \IBM\WBIConnect\router\was\bin
6. Type the following command to start the Document Manager:

   ```
   /startserver server1
   ```

   or

   ```
   startserver server1
   ```
7. When you see the message:

   ```
   Server server1 open for business
   ```

   log in to the Community Console, as described in the next section.

Logging in to the Community Console

The Community Console is the access point to WebSphere Business Integration Connect. Most of the tasks you will perform in setting up the hub require that you be logged in as the Hub Administrator (hubadmin), which is the super-user of the system.

Make sure you know the IP address of the computer on which the Console component is running. You will enter that address in the HTTP command.

1. From a browser, type the following URL:

   ```
   http://<IP_ADDRESS>:58080/console
   ```
2. Enter the following information:
   a. User Name: **hubadmin**
   b. Password: **Pa55word**

      Note: If you have already signed on to the Community Console and changed the default password of Pa55word, enter your new password in the Password field.
   c. Company Name: **Operator**

You see the Participant Search screen, which is always the first screen displayed when you log in to the Community Console.

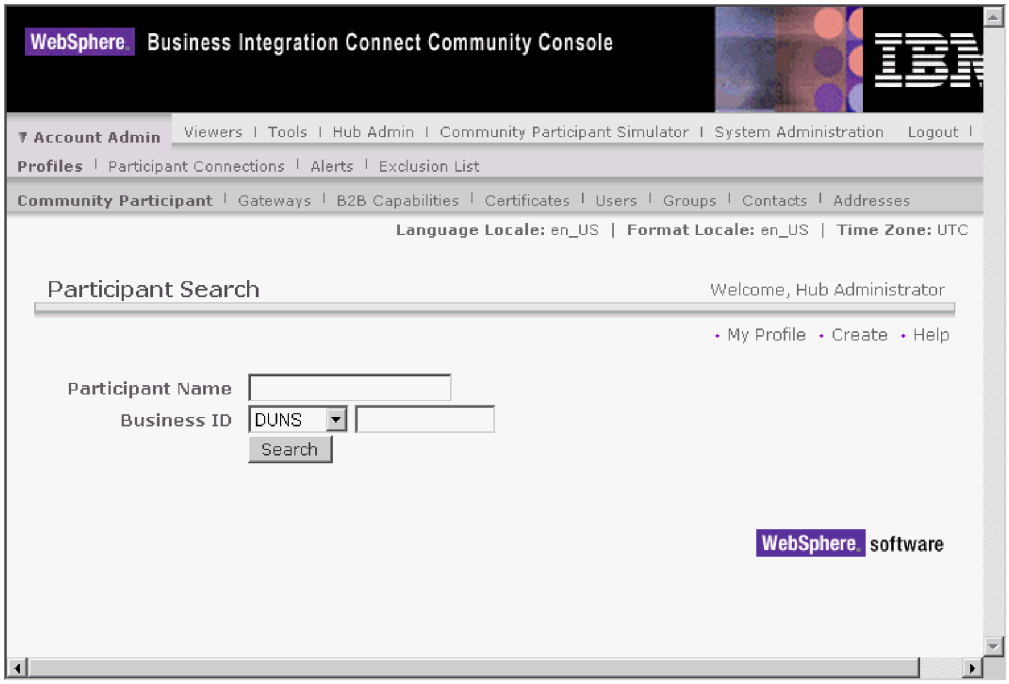


Figure 16. The Participant Search page

You will use this page later in the book to define participants.

If you click **Search** now, you will see that one participant, the Community Operator, is listed. The Community Operator is defined automatically by WebSphere Business Integration Connect.

Note: If you have not changed your password from the default of Pa55word, you will be prompted to do so before the Participant Search page is displayed.

Chapter 4. Configuring the Community Console

This chapter describes how to configure the Community Console so that you can control what participants see, how they log in to the console, and what access they have to various console tasks. Specifically, you can perform the following tasks:

- Change the default appearance of the console (for example, to include a company logo on the console)
- Set the password policy for participants when they log in to the console (for example, how many characters must be entered)
- Specify which elements of the console (for example, the Document Volume Report) are visible to participants

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

Specifying locale information and console branding

By default, the pages of the Community Console are presented in the English language. IBM might provide translations of the content in other languages as a set of files that can be uploaded. Other console items that might be provided by IBM for different locales are the logo and banner graphics, the stylesheet that is used to format the text on the screens, and the help system.

You can also choose to supply your own logo and banner to customize the Community Console. 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:

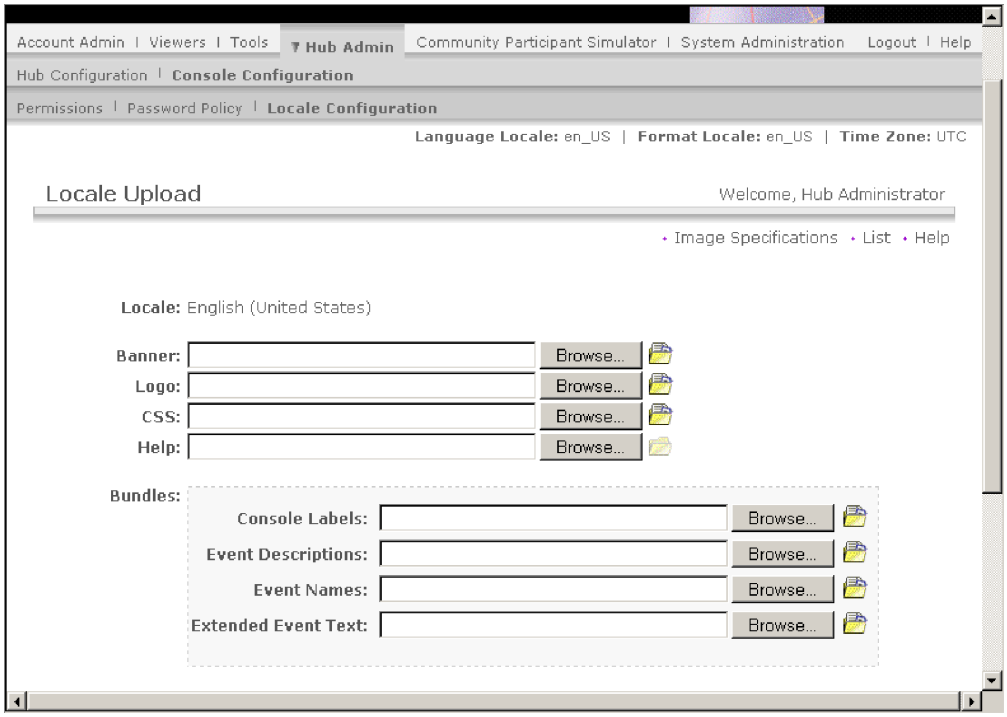


Figure 17. The Locale Upload page

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

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, so that they will 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 screen 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.

Localizing the data on the console

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
- Event Names, which contain text strings representing event names
- 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 participant 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 participant 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 screen is displayed.

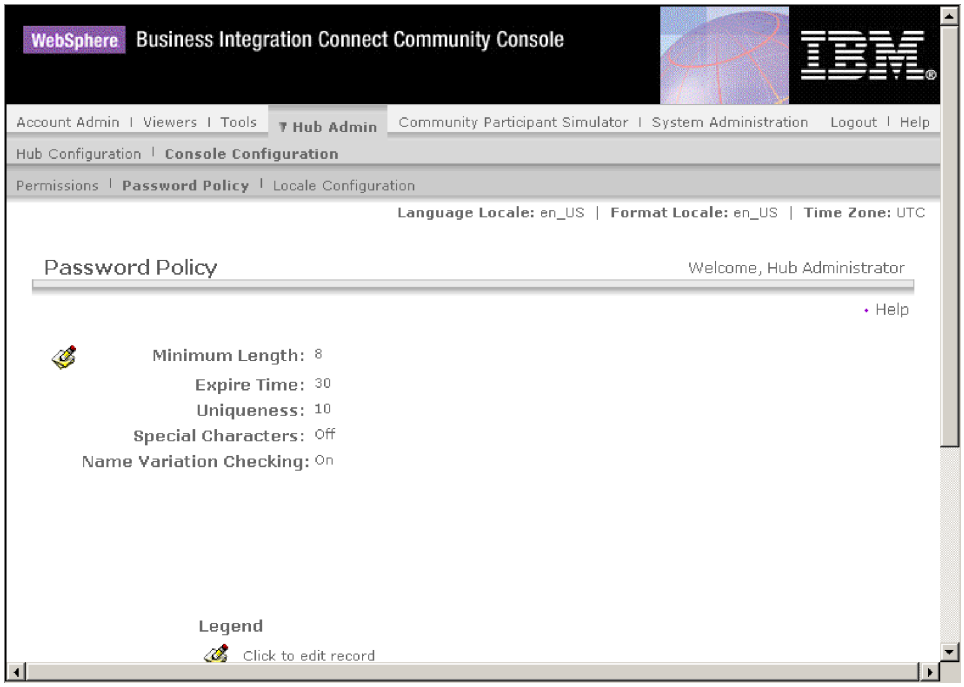


Figure 18. The Password Policy page

2. Click the  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 Community Operator, the Community Manager, and participants, have an Admin user. When you create a Community Manager or participant, you are actually creating the Admin user for that entity. (In the case of the Community Operator, the Hub Admin is automatically created, as is another Admin user for the hub.)

When you create the participant (as defined in “Creating participants” on page 49), you provide the participant with login information (such as the name to use to log in and the password). After the participant logs in, the participant creates additional users within the organization. The participant 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 participant would create a Volume group and add users to it.

Note: As the Hub Admin user, you can also define the users and groups for a participant.

The Admin user for the participant 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.

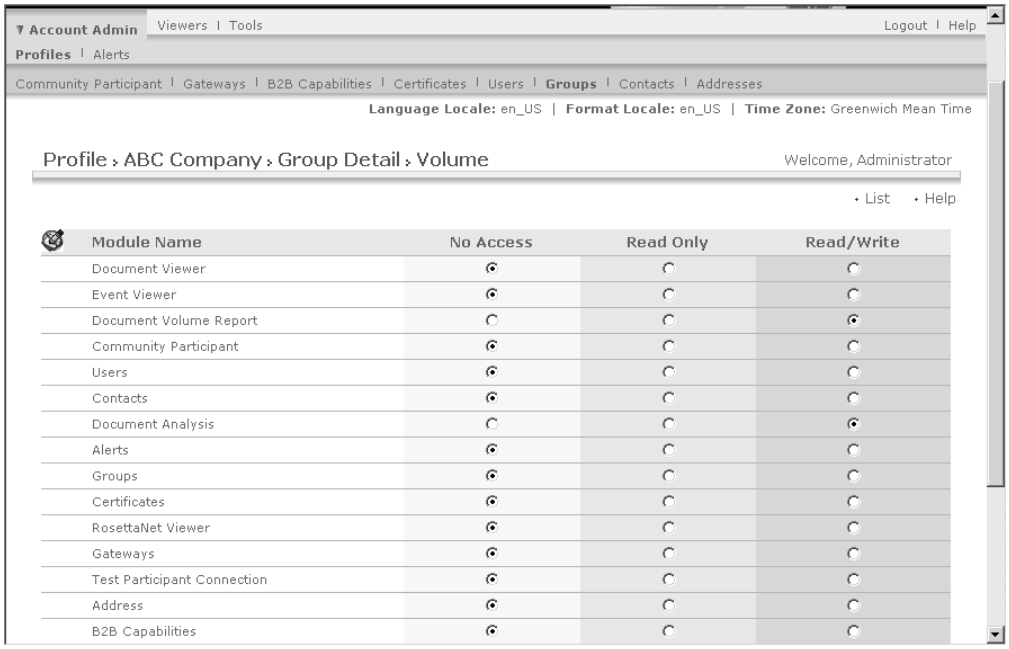


Figure 19. The Group Details page

The setting you, as the Hub Admin, 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 Admin), so even if you enable one of these modules for use by a participant, the module will not be displayed on the Group Details page for the participant.

Enabling or disabling permissions

From the Permission List screen, you can determine which permissions will be 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.

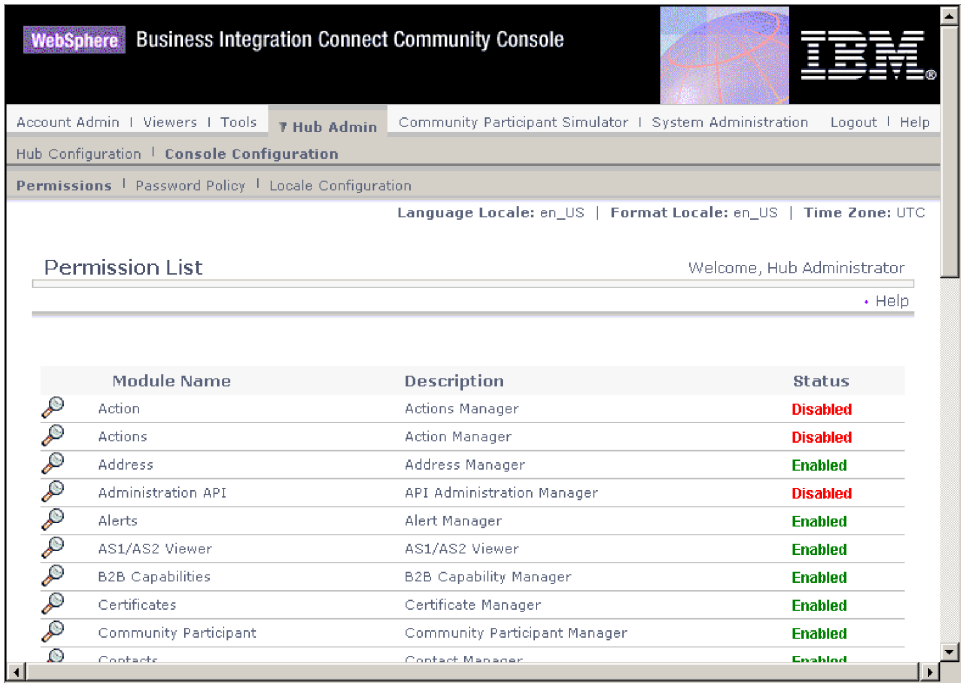


Figure 20. The Permission List page

2. View the default permissions to determine whether they are adequate for your hub community.
   - If the defaults are acceptable, click **Cancel**.
   - 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 **Yes**.

Chapter 5. Configuring the hub

WebSphere Business Integration Connect supports, by default, a set of transports as well as packages (such as AS2) and protocols (such as EDI-X12). You can add your own (user-created) transports for both targets and gateways. In addition, you can upload handlers to modify the way that the components process documents.

Uploading user-defined handlers

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 below).

Note: As mentioned in Chapter 1, "Introduction," you upload only user-defined handlers. The handlers supplied by WebSphere Business Integration Connect are already available.

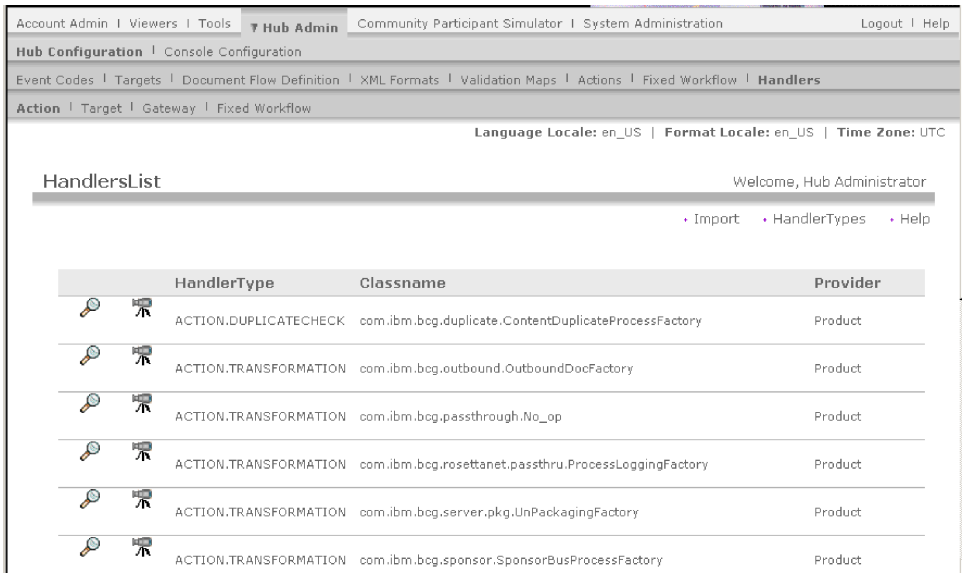
You can modify document flows by modifying fixed workflows, actions, targets and gateways. You modify these components by the handlers you associate with them.

Note: You can list the valid handler types for actions, targets, gateways, and fixed workflows by clicking **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, which can be **Action**, **Target**, **Gateway**, or **Fixed Workflow**.

The list of handlers currently defined for that particular component is displayed. For example, if you choose **Action**, you see the following page:



Notice that handlers provided by WebSphere Business Integration Connect are listed. They have a Provider ID of **Product**.

3. From the Handlers List page, click **Import**.
4. On the Import Handler page, specify the path to the XML file that represents the handler, or use **Browse** to search for that XML file.

After a handler is uploaded, you can use it to create new actions and workflows and to customize the configuration points of targets and gateways.

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

You cannot modify or delete handlers provided by WebSphere Business Integration Connect.

Setting up targets

Targets are the locations on the hub where documents are received. These documents can come from community participants (for eventual delivery to the Community Manager) or from the Community Manager (for eventual delivery to participants).

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

This illustration shows how four targets are set up to handle documents coming in to the hub. Two of the targets are for documents coming from participants, and two are for documents originating from the Community Manager. (Note that you can add transports to the list of those supported, by default, by WebSphere Business Integration Connect.)

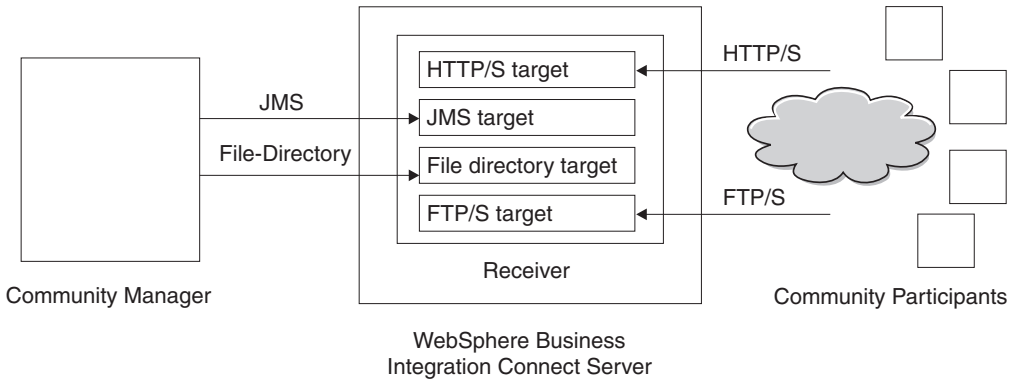


Figure 21. Transports and associated targets

To set up your targets, perform the following steps:

1. Click **Hub Admin > Hub Configuration > Targets**.
2. If you want to upload a user-defined transport, perform the following steps. Otherwise, go to 3.
   a. Click **Import Transport Type**.
   b. Enter the name of an XML file that defines the transport (or use **Browse** to navigate to the file).
   c. Click **Upload**.

   Note: From the Target List, you can also delete a user-defined transport type. You cannot delete a transport provided by WebSphere Business Integration Connect. Also, you cannot delete a user-defined transport after it has been used for creating a target.
3. Click **Create Target**.
4. Type a name for the target. For example, you might call the target HttpTarget. This is a required field. The name you enter here will be displayed on the Targets list.
5. Optionally indicate the status of the target. **Enabled** is the default. A target that is enabled is ready to accept documents. A target that is disabled cannot accept documents.
6. Optionally enter a description of the target.
7. Select a transport from the list. Note that, if you imported a user-defined transport, it appears on the list

The steps shown are common to all targets. After you select a target, however, additional fields are displayed on the page. The fields vary, depending on which transport you have chosen.

Here are the additional steps you take to configure the target, based on its transport type. After you provide the transport-specific information to define an HTTP/S or user-defined target, you can modify configuration points for the target. See “Modifying configuration points” on page 33.

Setting up an HTTP/S target

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

The following steps describe what you need to specify for an HTTP/S target:

1. Optionally, indicate the gateway type. The gateway 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 URI for the HTTP/S target. The name must begin with **bcgreceiver**. For example, you might enter bcgreceiver/submit. Documents coming into the server over HTTP/S would then be received at bcgreceiver/submit.
3. Optionally change the sync routing values:
   a. For **Max Sync Timeout**, enter the number of milliseconds a synchronous connection will remain open. The default is 600000.
   b. For **Max Sync Sim Conn**, enter the maximum number of synchronous connections the system will allow. The default is 100 (for the maximum number of simultaneous synchronous connections).
4. If you want to modify the configuration points, or if you are setting up a target for an AS2, cXML, RNIF, or SOAP document that will be involved in a synchronous exchange, see "Modifying configuration points" on page 33.

Setting up an FTP target

The following steps describe what you need to specify for an FTP target:

1. In the **FTP Route Directory** field, enter the root directory of the FTP server. Refer to "Configuring the FTP server for receiving documents" on page 13 for information on setting up the directory for an FTP server.
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 will retrieve the document for processing. The default value is 3 seconds.
3. Optionally enter a value for **Thread Nbr**, to indicate the number of documents the Document Manager will process simultaneously. The default value of 1 is recommended.
4. Optionally enter a value for **Exclude File Ext** 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. The default is that no file types are excluded.

Setting up an SMTP target

The following steps describe what you need to specify for an SMTP (POP3) target:

1. Optionally indicate the gateway type. The default is **Production**.
2. Enter the location of the POP3 server where mail is delivered.
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. Optionally enter a value for **Timeout**, to indicate the number of seconds the target will monitor the POP3 server for documents. This field is optional. The default is 1 ms.
6. Optionally enter a value for **Thread Nbr**, to indicate the number of documents the Document Manager will process simultaneously. The default value of 1 is recommended.
7. Optionally select the time of day (hours and minutes) when the SMTP target should poll the POP3 server for documents.

8. Optionally select the days of the week when polling should occur. The default is to poll on a daily basis.
9. Optionally select the days of the month when polling should occur. The default is to poll on daily basis.

Setting up a JMS target

The following steps describe what you need to specify for a JMS target:

1. Optionally indicate the gateway type. 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 Business Integration Connect for JMS, as described in "Configuring the hub for the JMS transport protocol" on page 16.
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.
5. Enter a value for the JMS factory name. This is a required field.
6. Optionally enter the Provider URL package.
7. Enter the JNDI factory name. If you do not enter anything, the value com.sun.jndi.fscontext.RefFSContextFactory is used. This is a required field.
8. Optionally enter a value for Timeout, to indicate the number of seconds the target will monitor the JMS queue for documents. This field is optional.
9. Optionally enter a value for **Thread Nbr**, 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 target to match the JMS configuration example in Chapter 2, you would:

1. Enter the value **JMSTarget** in the **Target Name** box.
2. Enter the value **file:/C:/TEMP/JMS/JMS** in the **JMS Provider URL** box
3. Enter the value **inQ** in the **JMS Queue Name** box.
4. Enter the value **WBICHub** in the **JMS Factory Name** box.

Setting up a File-system target

The following steps describe what you need to specify for a file-system target:

1. Optionally indicate the gateway type. The default is **Production**.
2. Enter a value for **Document Root Path** to indicate the directory where the documents will be received.
3. 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.
4. Optionally, enter a value for **File Unchanged Interval** to indicate the number of seconds the file size must remain unchanged before the Document Manager will retrieve the document for processing. The default value is 3 seconds.
5. Optionally enter a value for **Thread Nbr**, to indicate the number of documents the Document Manager will process simultaneously. The default value of 1 is recommended.

Modifying configuration points

For certain business protocols (RosettaNet, cXML, SOAP, and AS2) that will be involved in synchronous exchanges, you must specify a handler for the SyncCheck

configuration point. You can also modify the way an HTTP/S or user-defined target processes documents by applying an uploaded user-defined handler (or a system-supplied process) to other configuration points of the target.

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 29. You can also use a system-supplied handler, which is already available and does not have to be uploaded.

To modify the configuration points, perform the following steps:

1. If you are in the process of creating a target, continue to step 2. If you are updating a target configuration, click **Hub Admin > Hub Configuration > Targets**. Then click the magnifying glass icon next to the target. Finally, click  .
2. If you are specifying a handler for AS2, cXML, SOAP, or RNIF synchronous transactions, perform the following steps:
   a. Select **SyncCheck** from the **Configuration Point Handlers** list.
   b. Add the appropriate handler to the **Configured List** by selecting the handler from the **Available List** and clicking **Add**.

      Repeat this step if you want to add other handlers to the list. Remember that for targets, 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. 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.
   c. If you are finished defining handlers for this target, click **Save**. Otherwise, go on to step 3.
3. Select from the **Configuration Point Handlers** list the configuration point to be modified. The configuration points that can be modified for targets are **Preprocess**, **SyncCheck**, and **Postprocess**.

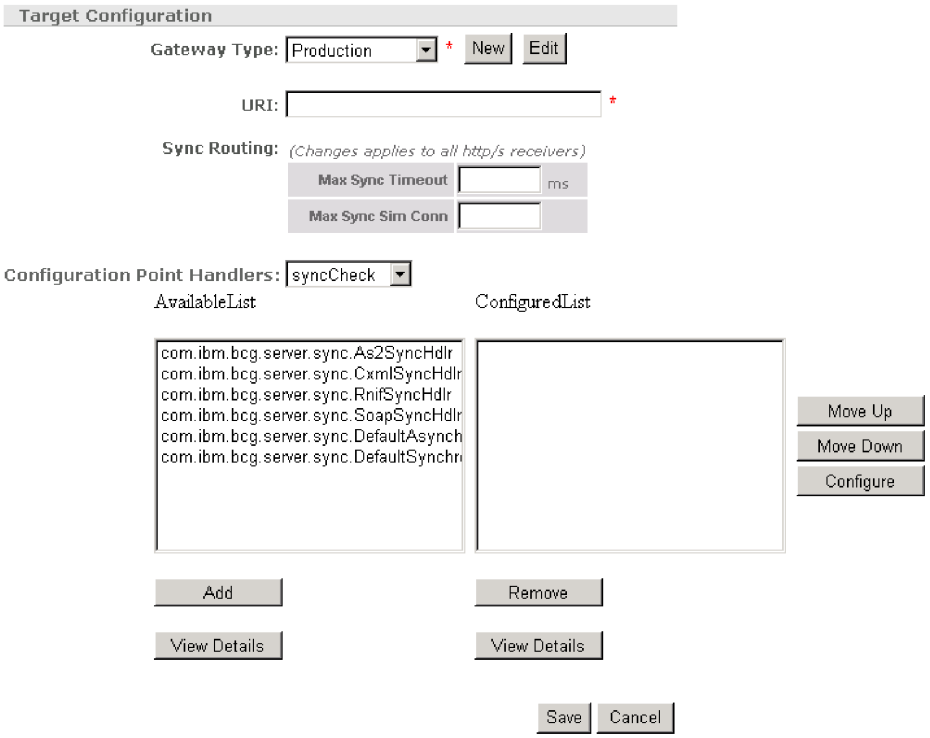


Figure 22. Target configuration point handlers

4. Perform one or more of the following steps for each handler you want to modify.
   a. Add a handler by selecting the handler from the **Available Handlers** list and clicking **Add**. The handler is moved to the **Configured Handlers** list.
   b. Remove a handler by selecting the handler from the **Configured Handlers** list and clicking **Remove**. The handler is moved to the **Available Handlers** list.
   c. Rearrange the order in which the handler is used 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**.
   e. Configure the handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured will be displayed.
5. Click **Save**.

Defining document flows and interactions

After you have created all the targets you need to receive documents from community participants and the Community Manager, the next step is to specify the types of documents you expect to receive at the hub. You do this from the Manage Document Flow Definition page.

A document flow definition is made up of, at minimum, a package, a protocol, and a document flow. For some protocols, an activity, action, and signal can be specified.

Using system-supplied packages and protocols

When you install WebSphere Business Integration Connect, a set of default packages (AS, None, Backend Integration) is displayed on this page. All the default packages are enabled (by default) for use. When you expand the packages, you see the choice of protocols that can be used with that package.

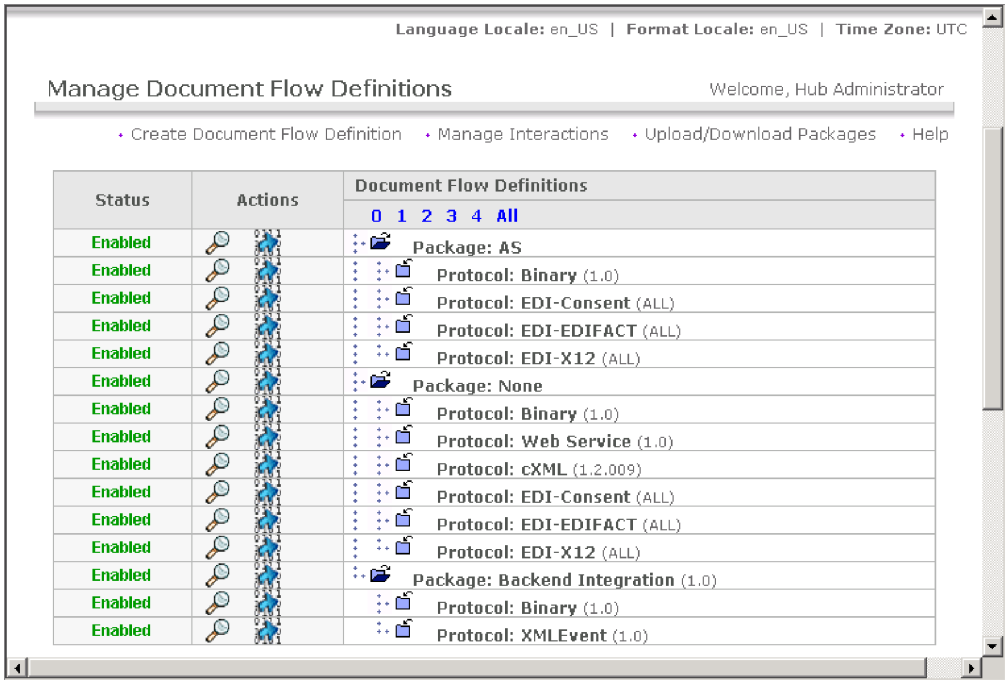


Figure 23. The default packages

For example, under **AS**, you see **EDI-X12**. If you selected **EDI-X12** under **AS**, WebSphere Business Integration Connect would be able to send or receive EDI-X12 documents wrapped in AS2 packaging. If you selected **None** and then **Web Service**, WebSphere Business Integration Connect would be able to request a Web service of a participant or provide a Web service to a participant.

If you are going to send or receive documents with the Web Service protocol, you must upload the WSDL file that is associated with the Web service, which is described in "Uploading packages" For more detailed information about using Web services, see Appendix C.

With the exception of Web services, if your hub community will use only these combinations of packages and protocols, you can skip ahead to "Creating interactions" on page 46. However, if you want to use a package or protocol that is not provided on the Manage Document Flow Definitions page, or if you want to support Web services, follow the procedures in the remainder of this section. Also, if you want to modify the inbound or outbound workflow steps or create or modify actions, see "Configuring document processing" on page 38.

Uploading packages

Business Integration Connect provides a way to import predefined RNIF Document Flow Definitions and WSDL files. RNIF Document Flow Definitions are uploaded in ZIP archives called packages. WSDL files can be uploaded

individually or together in a ZIP archive. If you are not exchanging RosettaNet documents or supporting Web services, skip this section and go on to “Configuring document processing” on page 38.

Uploading WSDL packages

This section describes how to upload a WSDL package associated with a Web service. See Appendix C for more complete information about using Web services with WebSphere Business Integration Connect.

To upload a WSDL package, perform the following steps:

1. Click **Hub Admin >Hub Configuration > Document Flow Definition**.
2. Click **Upload/Download Packages**.

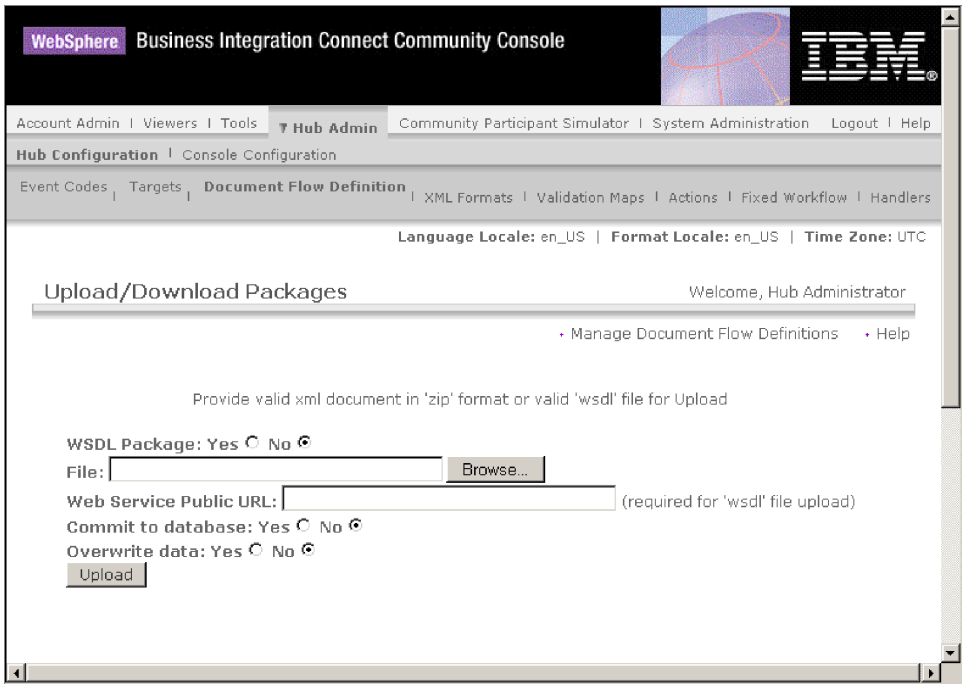


Figure 24. The Upload/Download Packages page

3. Select **Yes** for **WSDL Package**.
4. For **Web Service Public URL**, enter the public URL of the Web service provided by the Community Manager to the participant or by the participant to the Community Manager.
   - For a Web service provided by the Community Manager (which will be invoked by a participant), enter:

     `http(s)://<target host:port>/bcgreceiver/Receiver`

     The URL is typically the same as the production HTTP target.
   - For a Web service provided by a participant (which will be invoked by the Community Manager), enter the public URL of the participant with a query string. For example:

     `http(s)://<target host:port>/bcgreceiver/Receiver?to=<participant business ID>`
5. Click **Browse** and select the WSDL file.
6. Make sure **Commit to Database** is set to **Yes**.
7. Click **Upload**.

   The WSDL file is installed into the system.

Uploading RNIF packages

This section describes how to upload an RNIF package to be used to send and receive RosettaNet documents. See Appendix B for more complete information about using RosettaNet documents with WebSphere Business Integration Connect.

To upload an RNIF package:

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click **Upload/Download Packages**.
3. Select **No** for **WSDL Package**.
4. Click **Browse** and select the RNIF package.

   Note: The file in the ZIP archive must be within a directory titled Packages (for example: Packages/AS1.xml).
5. Make sure **Commit to Database** is set to **Yes**.
6. Click **Upload**.

   The package is installed into the system.

Configuring document processing

As described in Chapter 1, "Introduction," you can modify the system-supplied behavior for workflow steps by adding handlers to the steps. You can also modify the actions performed on a document by configuring handlers for the action. You can also create new actions.

This section describes how to add handlers for workflows and how to configure and create actions.

Configuring fixed workflows

Chapter 1, "Introduction" described that there are two fixed inbound workflow steps--one for unpackaging a protocol and one for parsing the protocol. For outbound workflows, there is one step, for protocol packaging.

WebSphere Business Integration Connect supplies a set of steps for each type of workflow.

If you are going to use a user-defined handler to configure a workflow step, upload the handler, as described in "Uploading user-defined handlers" on page 29.

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 magnifying glass 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.
4. Click the edit icon to edit the list of handlers.
5. Perform one or more of the following steps for each handler you want to modify.
   a. Add a handler by selecting the handler from the **Available Handlers** list and clicking **Add**. (A handler would appear in the **Available List** if you had uploaded a user-defined handler or if you had previously removed a handler from the **Configured Handlers** list.) The handler is moved to the **Configured Handlers** list.

b. Remove a handler by selecting the handler from the **Configured Handlers** list and clicking **Remove**. The handler is moved to the **Available Handlers** list.

c. Rearrange the order in which the handlers are called by selecting the handler and clicking **Move Up** or **Move Down**.

   Remember that handlers are called in the order in which they are listed in the **Configured Handlers** list. The first available handler that can process the request is the one that handles the request.

d. Cause a handler to be processed more than once by selecting it and then clicking **Repeat**.

6. Click **Save**.

Configuring actions

Chapter 1, "Introduction" described that actions can be made up of one or more steps. WebSphere Business Integration Connect 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 40.

Note: You cannot modify the actions supplied by WebSphere Business Integration Connect, although you can copy one of those actions and modify it, as described in "Creating actions" on page 40.

If you are going to use a user-defined handler to configure an action, upload the handler, as described in "Uploading user-defined handlers" on page 29.

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

1. Click **Hub Admin > Hub Configuration >Action**.
2. Click the magnifying glass 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 handler (action step) by selecting the handler from the **Available Handlers** list and clicking **Add**. (A handler would appear in the **Available List** if you had uploaded a user-defined handler or if you had previously removed a handler from the **Configured Handlers** list.) The handler is moved to the **Configured Handlers** list.

   b. Remove a handler by selecting the handler from the **Configured Handlers** list and clicking **Remove**. The handler is moved to the **Available Handlers** 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 Handlers** list.

   e. Configure the handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured will be 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

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 handler that will be invoked as part of the action, add the handler by selecting it from the **Available Handlers** list and clicking **Add**. (Any action handlers you uploaded appear in the **Available List**.) The handler is moved to the **Configured Handlers** list.

   Remember that handlers are called by the action in the order in which they appear in the **Configured List**, so 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 the handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured will be displayed.
8. Click **Save**.

Copying an action

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.

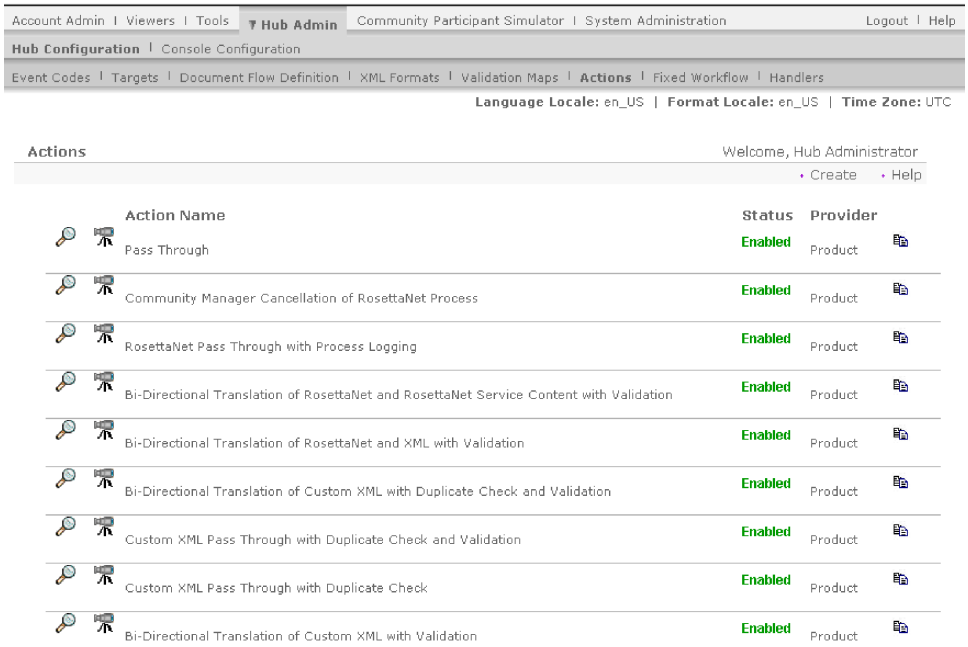


Figure 25. The Actions page

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. Perform one or more of the following steps for each handler you want to modify.
   a. Add a handler by selecting the handler from the **Available Handlers** list and clicking **Add**. (A handler would appear in the **Available List** if you had uploaded a user-defined handler or if you had previously removed a handler from the **Configured Handlers** list.) The handler is moved to the **Configured Handlers** list.
   b. Remove a handler by selecting the handler from the **Configured Handlers** list and clicking **Remove**. The handler is moved to the **Available Handlers** 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 Handlers** list.
   d. Configure the handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured will be displayed.
7. Click **Save**.

Managing custom XML

Perform the steps in this section only if you will be using a custom XML format.

XML (Extensible Markup Language) is the universal format for structured documents and data on the Web. Using the Manage XML Protocols page, you can create and manage custom XML formats that can be added to the list of available Document Flow Definitions.

An XML format defines the paths within a set of XML documents. This enables 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.

Creating an XML format is a multi-step process. You must:

1. Create a protocol for the format and associate it with a package or packages
2. Create a document flow for the format and associate it with the newly created protocol
3. Create the format

You then create a valid interaction for the newly created format.

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 87.

Creating a CustomXML protocol definition format

The following steps describe how to create a custom XML protocol definition format:

1. Click **Hub Admin > Document Flow Definitions > Create Document Flow Definition**.

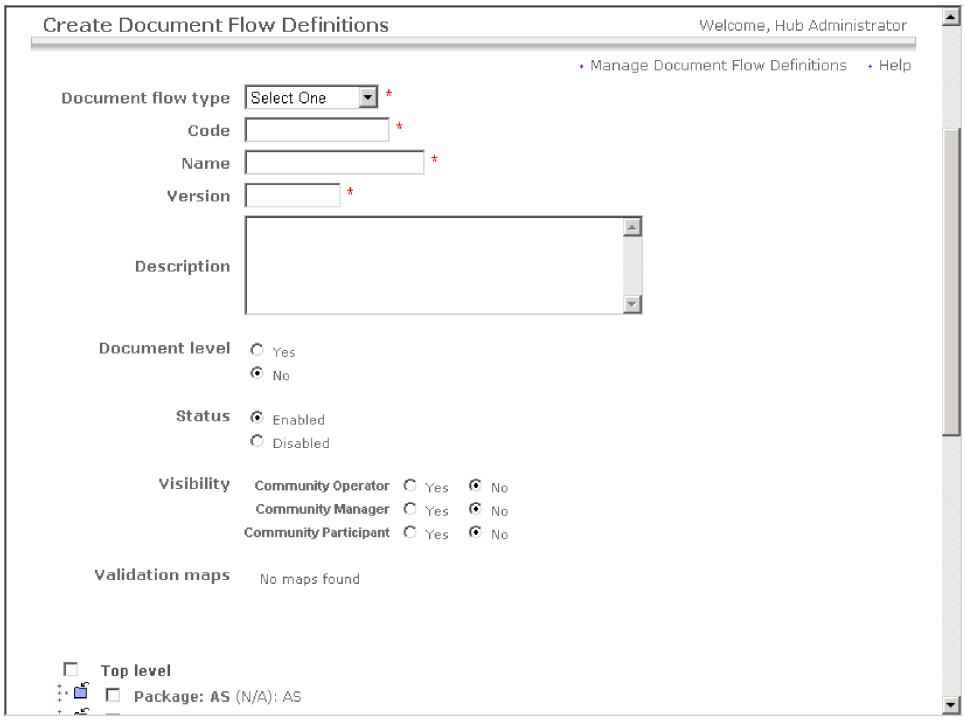


Figure 26. Create Document Flow Definitions page

2. For **Document flow type**, select **Protocol**.
3. For **Code**, enter the value for the type of object you selected in the previous step. For example, you might want to enter XML.
4. For **Name**, enter an identifier for the document flow definition. For example, for a custom XML protocol, you could enter Custom_XML. This field is required.

5. For **Version**, enter **1.0**.
6. Enter an optional description of the protocol.
7. Set **Document Level** to **No**, because you are defining a protocol, rather than a document flow (which you will define in the next section).
8. Set **Status** to **Enabled**.
9. Set **Visibility** for this protocol. You will probably want it to be visible to all participants.
10. Select the packages in which this new protocol will be wrapped. For example, if you want this protocol to be associated with all three packages, select **Package: AS**, **Package: None**, and **Package: Backend Integration**.
11. Click **Save**.

Creating a document definition flow

Next, use the Create Document Flow Definition page again to create a document flow.

1. Click **Hub Admin > Document Flow Definitions > Create Document Flow Definition**.
2. For **Document flow type**, select **Document Flow**.
3. For **Code**, enter the value for the type of object (document flow) you selected in the previous step.
4. For **Name**, enter an identifier for the document flow definition. For example, you could enter XML_Tester as a name for the document flow. This field is required.
5. For **Version**, enter **1.0**.
6. Enter an optional description of the protocol.
7. Set **Document Level** to **Yes** (because you are defining a document level).
8. Set **Status** to **Enabled**.
9. Set **Visibility** for this flow. You will probably want it to be visible to all participants.
10. Click the folder icon to expand each package you selected in the previous procedure. Expand the folder and select the name of the protocol you created in the previous section (for example, Protocol: CustomXML.).
11. Click **Save**.

The following is an example of what the AS Package portion of the Manage Document Flow Definitions page would look like if you created a protocol of CustomXML, associated the protocol with AS, None, and Backend Integration packaging, and created a document flow of XML_Tester:

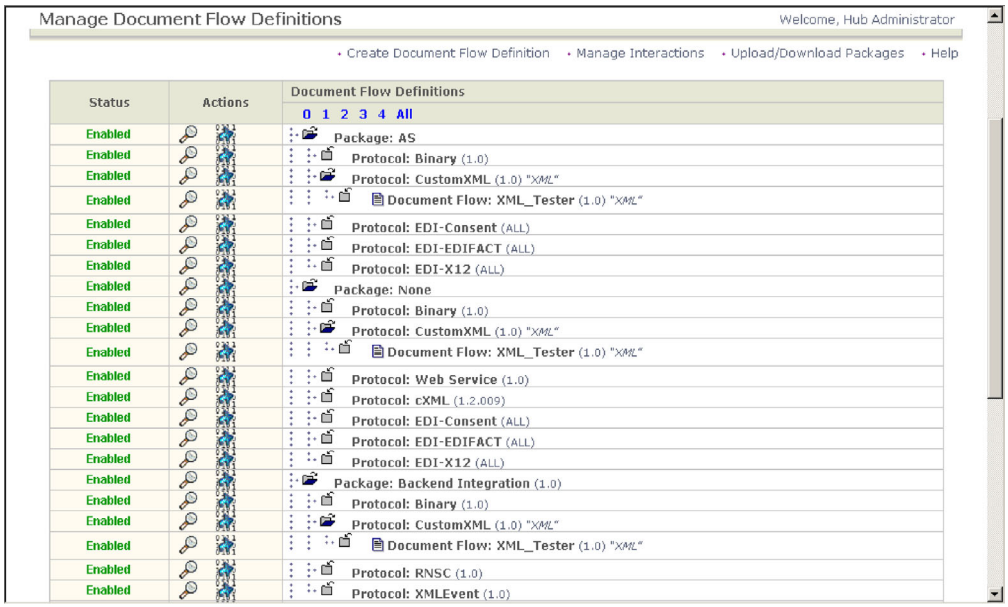


Figure 27. Document Flow Definition page with new Custom XML protocol and document flow added

Creating an XML format

After you create a custom XML protocol (and associate it with a package or set of packages) and create an associated document flow, you are ready to create the XML format.

To create an XML format, use the following procedure.

1. Click **Hub Admin > Hub Configuration > XML Formats**.
2. Click **Create XML Format**.

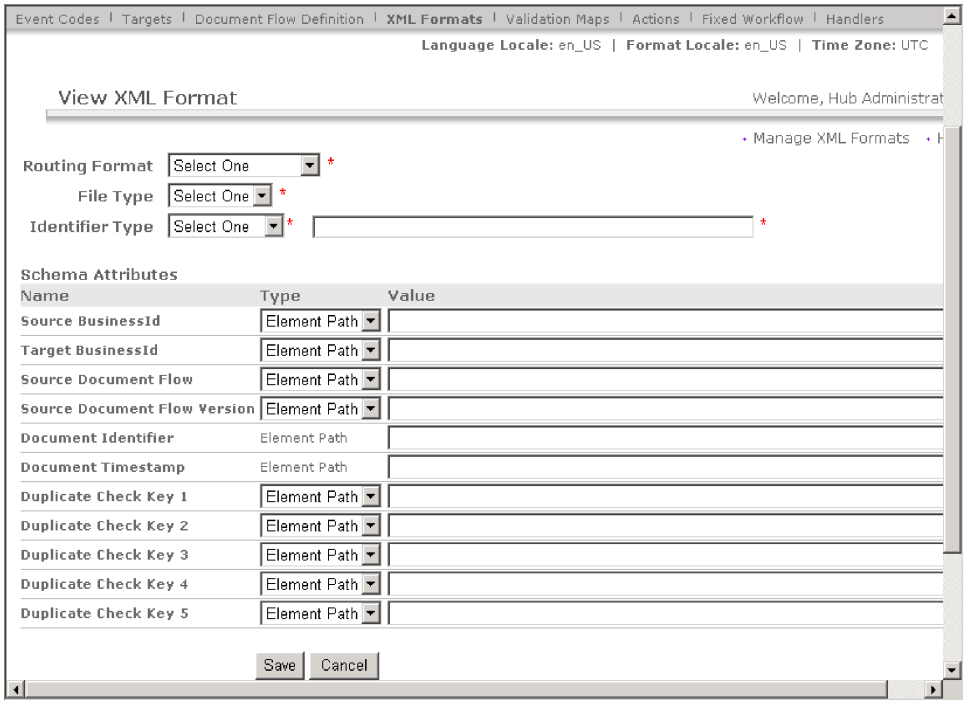


Figure 28. The View XML Format page

3. For **Routing Format**, select the document flow definition with which this format will be associated.
4. For **File Type**, select **XML**.

   Note: XML is the only option available for file type.
5. For **Identifier Type**, select the element used to identify the incoming document type. The choices are **DTD**, **Name Space**, or **Root Tag**.
6. For each field for which a choice of types is offered, select either **Element Path**, which is the path to the value in the document, or **Constant**, which is the actual value in the document. Then provide a value.
   a. For **Source/Target Business ID**, enter the path of the business ID. This field is required.
   b. For **Source Document Flow & Version**, enter an expression that defines the path to the Document Flow and Version value within the XML document. This field is required.
   c. For **Document Identifier**, enter the path for the document ID number.
   d. For **Document Timestamp**, enter the path for the document creation time stamp.
   e. For **Duplicate Check Key 1-5**, enter paths used to identify the routing of a duplicate document.
7. Click **Save**.

Using validation maps

WebSphere Business Integration Connect uses validation maps to validate the structure of RosettaNet or XML documents. If you have no need to import validation maps, skip ahead to “Creating interactions” on page 46.

Adding validation maps

An action can have an associated validation map to ensure that the destination participant 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 Flow 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 Business Integration Connect can read files
2. Click **Hub Admin > Hub Configuration > Validation Maps**.
3. Click **Create**.
4. Type a description of the validation map. Choose the path and name of the schema file you want to use to validate documents.
5. Click **Save**.

Associating maps with Document Flow Definitions

To associate a validation map with a Document Flow Definition, use the following procedure.

1. Click **Hub Admin > Hub Configuration > Validation Maps**. The Console displays the Manage Maps page.
2. Click the magnifying glass icon next to the validation map you want to associate with the Document Flow Definition.
3. Click the folder icon to individually expand to the **Action** level, or select **All** to expand the entire tree.
4. Select the Document Flow Definition you want associated with the validation map.
5. Click **Submit**.

Creating interactions

After you have defined all the document flows you want to use at the hub, you create interactions. Interactions define the possible combinations of document flows that the hub will support.

To create interactions, use the following procedure.

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click **Manage Interactions**.
3. Click **Create Interaction**.

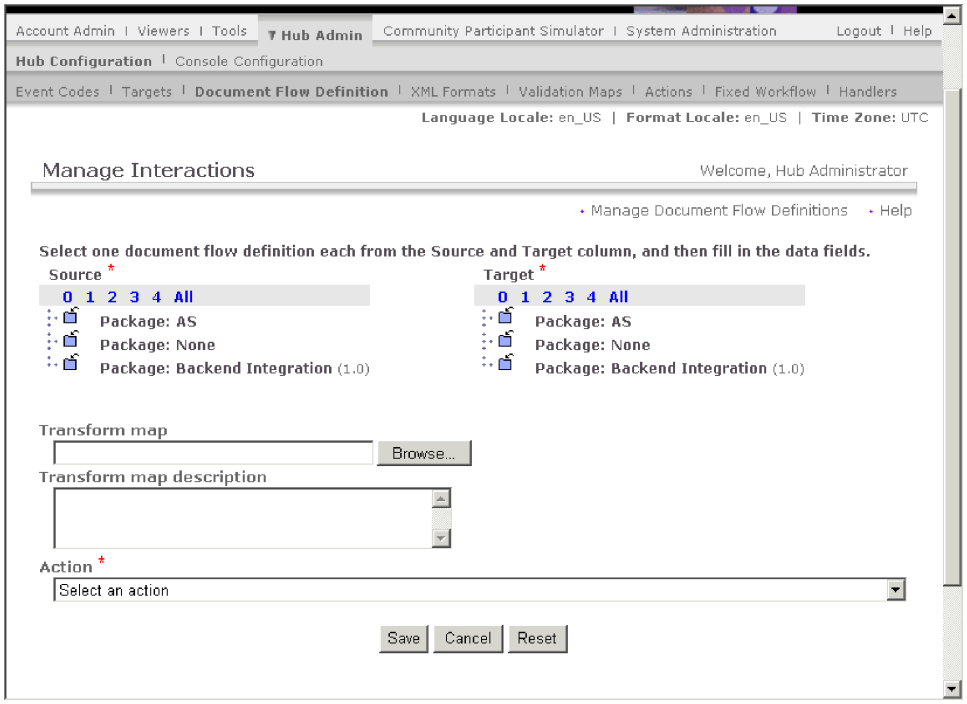


Figure 29. Manage Interactions page

The Manage Interactions page contains all the possible combinations of Package, Protocol and Document Flows, both those supplied by the system and those that you uploaded or created.

4. In the **Source** tree, click the folder icon to individually expand a node to the appropriate **Document Flow Definition** level or select **All** to expand the entire tree.
5. Select the Document Flow Definition you want as the source of the interaction.
6. In the **Target Document Flow Definition** tree, click the folder icon to individually expand a node to the appropriate **Document Flow Definition** level or select **All** to expand the entire tree.
7. Select the Document Flow Definition you want as the destination of the interaction.
8. If you need to translate data from one protocol to another, in the **Transform Map Document** field, type the name of the transformation map file or click **Browse** to navigate to the file.
9. Optionally, in the **Transform Map Description** field, type a description.
10. In the **Action** field, select the action that WebSphere Business Integration Connect is to perform in this interaction. Note that any actions you have created are listed.
11. Click **Save**.

Summary

In this chapter, you have configured the hub and are now ready to define participants, establish B2B capabilities, and define connections between participants and the Community Manager. You have learned how to perform the following tasks:

- Define targets for all the transports by which documents will arrive at the hub

- Upload WSDL or RNIF packages to add to the Document Flow Definition list, if needed
- Customize the processing of documents by configuring the fixed workflow steps and actions or creating actions, if needed
- Create custom XML formats to add to the Document Flow Definition List, if needed
- Upload transformation maps and associate them with document flows, if needed
- Create interactions to designate which combinations of exchanges are possible

Chapter 6. Creating participants and participant connections

After you have set up the hub, including establishing targets and setting up document flow definitions and interactions, you are ready to create the participants for your hub community. After creating the participants, you establish their B2B capabilities and then create connections between the participants and the Community Manager.

Creating participants

To create a participant, you need to know, at minimum, the following information about the participant:

- The IP address of the participant
- The Business ID that the participant 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 participant chooses to use to identify the company

For each participant (including the Community Manager) you want to add to the hub community, follow this procedure:

1. Click **Account Admin > Profiles > Community Participant**.
2. Click **Create**.
3. Enter the name the participant will use when logging in to the hub.
4. Enter the company name or some other descriptive name for the participant.
5. Select the type of participant. Note that WebSphere Business Integration Connect supports only one Community Manager and one Community Operator. If you are setting up the Community Manager, select **Community Manager**. Otherwise, select **Community Participant**.
6. Select the status for the participant. When you are creating a participant, you will probably want to use the default value of **Enabled**.
7. Optionally enter the type of company in the **Vendor** field.
8. Optionally enter the Web site of the participant.
9. Click **New** under **Business ID**.
10. Specify a type from the list, and enter the appropriate identifier. WebSphere Business Integration Connect uses the number you enter here to route the document to and from the participant.

    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 participant. In some cases, more than one Business ID is required. For example, when the hub sends and receives EDI-X12 or EDIFACT documents, it uses both the DUNS and Freeform IDs during the document exchange.

The Freeform ID is formed by inserting a hyphen (-) between the second and third digits of the DUNS. For example, if the DUNS ID is 810810810, the required Freeform would be 81-0810810. Both the Community Manager and the participants involved in these types of document flows should have both a DUNS and Freeform ID.

11. Optionally enter an IP address for the participant by performing the following steps:
    a. Under **IP Address**, click **New**.
    b. Specify the gateway type.
    c. Enter the IP address of the participant.
12. Click **Save**.

When you create a participant, you are actually creating the Admin user for that participant. Admin users can then create individual users within their organizations, or, as Hub Admin, you can create the users for the participants.

Setting up gateways for the participants

WebSphere Business Integration Connect uses gateways to route documents to their proper destination. The outbound transport protocol determines which information is used during gateway configuration.

Transports supported (by default) for participant gateways include the following:

- HTTP/1.1
- HTTPS/1.0
- HTTPS/1.1
- FTP
- FTPS
- JMS
- SMTP
- File directory

You can also specify a user-defined transport, which you upload during the creation of the gateway.

As the Hub Admin, you can set up the gateways for your participants, or the participants can perform this task themselves. In this chapter, you will see how to perform the task for the participants.

Creating gateways

To create gateways, use the following procedure.

1. Click **Account Admin > Profiles > Community Participant**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all participants.
3. Click the magnifying glass icon to display the participant's profile.
4. Click **Gateways**.
5. Click **Create**. The Console displays the Gateway Detail screen.
6. If you want to upload a user-defined transport, perform the following steps. Otherwise, go to step 7.
    a. Click **Import Transport Type**.

b. Enter the name of an XML file that defines the transport (or use **Browse** to navigate to the file).
c. Click **Upload**.

Note: From the Gateway List, you can also delete a user-defined transport type. You cannot delete a transport provided by WebSphere Business Integration Connect. Also, you cannot delete a user-defined transport after it has been used for creating a gateway.

7. Click **Create**.
8. Type a name to identify the gateway. This is a required field.
9. Optionally indicate the status of the gateway. **Enabled** is the default. A gateway that is enabled is ready to send documents. A gateway that is disabled cannot send documents.
10. Optionally indicate whether the gateway is Online or Offline. The default is **Online**.
11. Optionally enter a description of the gateway.

The steps shown are common to all gateways. After you select a gateway, however, the choices on the screen vary. Here are the additional steps you take to configure the gateway, based on its transport type.

Note that, after you provide the transport-specific information to define a gateway, you can also modify configuration points for the gateway.

Creating an HTTP gateway

To create an HTTP gateway:

1. In the **Target URI** field, enter the URI where the document will be delivered. This field is required.

   The format is: http://<servername>:<optional port>/<path>

   An example of this format is:

   ```
   http://anotherwbicserver.ibm.com:57080/bcgreceiver/Receiver
   ```

2. Optionally enter a user name and password, if a user name and password are required to access the secure HTTP server.
3. In the **Retry Count** field, enter the number of times you want the gateway 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 gateway 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 gateway 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 gateway 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 to configure the preprocess or postprocess step for the gateway, go to "Modifying configuration points for gateways" on page 56. Otherwise, click **Save**.

Creating an HTTPS gateway

To create an HTTPS gateway:

1. In the **Target URI** field, enter the URI where the document will be delivered. This field is required.

   The format is: https://<servername>:<optional port>/<path>

   For example:

   `https://anotherwbicserver.ibm.com:57443/bcgreceiver/Receiver`
2. Optionally enter a user name and password, if a user name and password are required to access the secure HTTP server.
3. In the **Retry Count** field, enter the number of times you want the gateway 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 gateway 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 **Validate Client SSL Cert** field, select **Yes** if you want the digital certificate of the sending partner to be validated against the DUNS number associated with the document. The default is **No**.
8. In the **Auto Queue** field, select **Yes** if you want the gateway 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 gateway 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 gateway, go to "Modifying configuration points for gateways" on page 56. Otherwise, click **Save**.

Creating an FTP gateway

To create an FTP gateway:

1. In the **Target URI** field, enter the URI where the document will be delivered. This field is required.

   The format is: ftp://<ftp servername>: <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 FTP server.
3. In the **Retry Count** field, enter the number of times you want the gateway 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 gateway 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 gateway 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 gateway 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 to configure the preprocess or postprocess step for the gateway, go to "Modifying configuration points for gateways" on page 56. Otherwise, click **Save**.

Creating an SMTP gateway

To create an SMTP gateway:

1. In the **Target URI** field, enter the URI where the document will be delivered. This field is required.

   The format is: mailto:<user@servername>

   For example:

   ```
   mailto:admin@anotherwbicserver.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 gateway 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 gateway 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 gateway 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 gateway 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 **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 gateway, go to "Modifying configuration points for gateways" on page 56. Otherwise, click **Save**.

Creating a JMS gateway

To create a JMS gateway:

1. In the **Target URI** field, enter the URI where the document will be delivered. This field is required.

For WebSphere MQ JMS, the format of the target URI is as follows:

```
file:///<user_defined_MQ_JNDI_bindings_path>
```

For example:

```
file:///opt/JNDI-Directory
```

The directory contains the ″ .bindings″ file for the file-based JNDI. This file indicates to WebSphere Business Integration Connect how to route the document to its intended destination.

For participant gateways, the participant will probably provide the ".bindings" file. Internal JMS gateways (that is, the Community Manager gateway) can be produced using JMSAdmin as discussed in Chapter 2, "Preparing to configure the hub."

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 gateway 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 gateway 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 gateway 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 gateway 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.
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. This is an optional field.
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.
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 as described in Chapter 2, "Preparing to configure the hub."
15. In the **Connection Timeout** field, enter the number of seconds a socket will remain open with no traffic. The default is 120 seconds.

16. If you want to configure the preprocess or postprocess step for the gateway, go to "Modifying configuration points for gateways" on page 56. Otherwise, click **Save**.

Creating a file-directory gateway

To create a file-directory gateway:

1. In the **Target URI** field, enter the URI where the document will be delivered. This field is required.

   The format for UNIX systems and for Windows systems in which the file directory is on the same drive on which WebSphere Business Integration Connect is installed is: file:///<path to target directory>

   For example:

   ```
   file:///localfiledir
   ```

   where *localfiledir* is a directory off the root directory.

   For Windows systems in which the file directory is on a separate drive from WebSphere Business Integration Connect, the format is: file:///<drive letter>:/<path>
2. In the **Retry Count** field, enter the number of times you want the gateway 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 gateway 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 gateway 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 gateway is placed online manually.
7. If you want to configure the preprocess or postprocess step for the gateway, go to "Modifying configuration points for gateways" on page 56. Otherwise, click **Save**.

Creating an FTPS gateway

To create an FTPS gateway:

1. In the **Target URI** field, enter the URI where the document will be delivered. This field is required.

   The format is: ftp://<ftp servername>:<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 gateway 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 gateway 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 gateway 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 gateway is placed online manually.
8. If you want to configure the preprocess or postprocess step for the gateway, go to "Modifying configuration points for gateways." Otherwise, click **Save**.

Note: For an outbound FTPS gateway to work correctly, you need at least the CA certificate of the FTPS server loaded under the Hub Operator's profile as a Root certificate. (You use **Account Admin > Profile > Certificates** to load a certificate.) When you load this certificate, WebSphere Business Integration Connection will trust the certificate of the FTPS server.

If the FTPS server requires client authentication as well, you must have a client certificate loaded under the Hub Operator's profile as an SSL certificate. WebSphere Business Integration Connect provides this certificate to the FTPS server. The FTPS server of the participant must be set up to trust your certificate.

For more information about security, see Chapter 7, "Setting up security for inbound and outbound exchanges."

Modifying configuration points for gateways

As described in Chapter 1, "Introduction," you can modify two processing points for a gateway--Preprocess and Postprocess.

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 29. You can also use a system-supplied handler, which is already available and does not have to be uploaded.

To modify a configuration point:

1. If you are in the process of creating a gateway, continue to step 6. If you are updating a gateway configuration, click **Account Admin > Profiles > Community Participant**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all participants.
3. Click the magnifying glass icon to display the participant's profile.
4. Click **Gateways**.
5. Click the magnifying glass icon to display the gateway, and then click the edit icon to edit the gateway.
6. Select from the **Configuration Point Handlers** list the configuration point to be modified. The configuration points that can be modified for gateways are **Preprocess** and **Postprocess**.
7. Perform one or more of the following steps for each handler you want to modify.
   a. Add a handler by selecting the handler from the **Available Handlers** list and clicking **Add**. The handler is moved to the **Configured Handlers** list.

Note: WebSphere Business Integration Connect does not supply default gateway handlers. The only handlers in the **Available List** will be those that you uploaded.

b. Remove a handler by selecting the handler from the **Configured Handlers** list and clicking **Remove**. The handler is moved to the **Available Handlers** list.

c. Rearrange the order in which the handler is used by selecting the handler and clicking **Move Up** or **Move Down**.

d. Configure the handler by selecting it from the **Configured List** and clicking **Configure**. The list of attributes that can be configured will be displayed.

8. Click **Save**.

Setting up B2B capabilities

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

As the Hub Admin, you can set up the B2B capabilities of your participants, or the participants can perform this task themselves. In this chapter, you will see how to perform the task for the participants.

You use the B2B Capabilities feature to associate a participant's B2B capabilities with a Document Flow Definition.

Use the following procedure to set the B2B capabilities of each participant.

1. Click **Account Admin > Profiles > Community Participant**.
2. Enter search criteria and click **Search**, or click **Search** without entering any search criteria to display a list of all participants.
3. Click the magnifying glass icon to display the participant's profile.
4. Click **B2B Capabilities**. The B2B capabilities screen is displayed. The right side of the screen shows the packages, protocols, and business processes supported by the system as Document Flow Definitions.

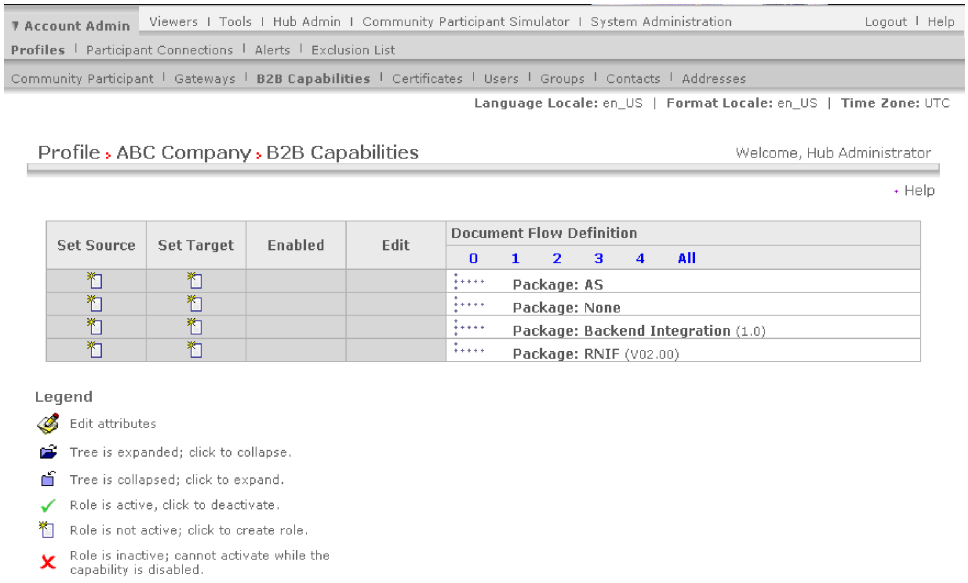


Figure 30. The B2B Capabilities page

5. Click the activate icon under the **Set Source** column for the Packages on the right that contain business processes you will send to the participants or Community Manager.
6. Select both if you will send and receive those same processes. The Console displays a check if the Document Flow 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 participant and the corresponding confirmation from another. This also applies to Set Target.
7. Click the folder icon at the **Package** level to expand an individual node to the appropriate Document Flow Definition level or select a number from **0-4** or **All** to expand all displayed Document Flow Definitions to the selected level.
8. Again, select the **Set Source**, **Set Target**, or both roles for the lower Protocol, Document Flow, Action, and Activity levels for each Document Flow Definition your system supports.

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

   If a package Document Flow Definition is disabled, all lower-level Document Flow Definitions in that same node are also disabled, regardless of whether their individual status was enabled. If a lower-level Document Flow Definition is disabled, all higher-level definitions within the same context remain enabled. When a Document Flow Definition is disabled, all preexisting connections and attributes continue to function. The disabled Document Flow Definition only restricts the creation of new connections.
10. Optionally click the edit icon if you want to edit any of the attributes of a protocol, package, document flow, 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**.

Activating participant connections

Participant connections contain the information necessary for the proper exchange of each document flow. A document cannot be routed unless a connection exists between the Community Manager and one of its participants.

The system automatically creates connections between the Community Manager and participants based on their B2B capabilities.

You search for these connections and then activate them.

When selecting a Source and a Target, observe the following guidelines:

- The Source and Target must be unique.
- Do not mix a production gateway with a test gateway when selecting Source and Target; otherwise, an error occurs.
- Both the Source and the Target must be production or test gateways.

Use the following procedure to perform a basic search for connections and then activate the connections.

1. Click **Account Admin > Participant Connections**. The Console displays the Manage Connections screen.
2. Under **Source**, select a Source.
3. Under **Target**, select a Target.

   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 Console displays the Manage Connections screen. This screen shows the package, protocol, and document flow for the source and target. It also provides buttons you can click to view and change partner-connection status and parameters.
6. Click **Attributes** if you want to view or change the attribute values.
7. Click **Actions** if you want to view or change an action.
8. Click **Gateways** if you want to view or change the source or target gateway.

Summary

In this chapter, you created the Community Manager and participants, specifying such information as the IP address and DUNS ID of the participants. After creating the participants, you established gateways for them to indicate where documents should be routed.

Next, you selected the B2B capabilities of the Community Manager and participants, indicating the packages, protocols, and document flows the Community Manager and participant could send and receive. Finally, you activated participant connections, based on the B2B capabilities of the Document Manager and participants.

Chapter 7. Setting up security for inbound and outbound exchanges

With WebSphere Business Integration Connect, you can install and use the following types of certificates for inbound and outbound transactions:

- Secure Sockets Layer (SSL), for server and client
- Digital signature
- Encryption

Understanding terms and concepts

This section provides a general overview of the types of security, the tools used to generate and upload certificates, and the types of data stores installed by WebSphere Business Integration Connect.

Types of security

This section gives a brief overview of SSL, digital signatures, and encryption.

SSL

WebSphere Business Integration Connect can use SSL to secure inbound and outbound documents. An inbound document is one that is sent to the hub. An outbound document is one that is sent from the hub.

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 authenticate each other's identify.

An SSL connection begins with a handshake. During this stage, the applications exchange digital certificates, agree on the encryption algorithms to use, and generate encryption keys used for the remainder of the session.

The SSL protocol provides the following security features:

- Server authentication, which means that the server uses its digital certificate to authenticate itself to clients
- Client authentication, an optional step in which clients might be required to authenticate themselves to the server by providing their own digital certificate

Digital signature

Digital signing is the mechanism for ensuring non-repudiation. Non-repudiation means that a participant cannot deny having originated and sent a message. It also ensures that the participant cannot deny having received a message.

A digital signature allows an originator to sign a message so that the originator is verified as the person who actually sent the message. It also ensures that the message has not been modified since it was signed.

Encryption

WebSphere Business Integration Connect uses a cryptographic system known as public key encryption to secure the communication between participants and the hub. Public key encryption uses a pair of mathematically related keys. A document

encrypted with the first key must be decrypted with the second, and a document encrypted with the second key must be decrypted with the first.

Each participant in a public key system has a pair of keys. One of the keys is kept secret; this is the private key. The other key is distributed to anyone who wants it; this is the public key. WebSphere Business Integration Connect uses a participant's public key to encrypt a document. The private key is used to decrypt a document.

The ikeyman utility

As described in the sections that follow, you use the IBM Key Management Tool (ikeyman) to create key databases, public and private key pairs, and certificate requests. You can also use ikeyman to create self-signed certificates. The ikeyman utility is included in the <WBIC_install_dir>/router/was/bin directory, which WebSphere Business Installation Connect creates during installation.

You can also use ikeyman to generate a request for a certificate to a Certifying Authority (CA).

Note: You can also use the createCert.sh utility to generate self-signed certificates.

Community Console

You use the Community Console to install all the required client, signature, and encryption certificates for WebSphere Business Integration Connect storage. You can also use the Community Console to install Root and CA (Certifying Authority) certificates.

Note: When a participant's certificate expires, it is the participant's responsibility to obtain a new certificate. The Community Console's Alert feature includes certificate expiration alerts for certificates stored in WebSphere Business Integration Connect.

Keystores and truststores

When you install WebSphere Business Integration Connect, a keystore and truststore for the Receiver and Console are installed.

- A keystore is a file that contains your public and private keys.
- A truststore is a key database file that contains the public keys for your participants' self-signed and CA certificates. The public key is stored as a signer certificate. For commercial CA, the CA root certificate is added. The truststore file can be a more publicly accessible key database file that contains all the trusted certificates.

By default, the two keystores and two truststores are created in the `WBIC_install_root/common/security/keystore` directory. The names are:

- receiver.jks
- receiverTrust.jks
- console.jks
- consoleTrust.jks

The default password for accessing all four stores is WebAS. The embedded WebSphere Application Server is configured to use these four stores.

Note: The following Unix command can be used to change the password of the keystore file:

```
/WBIC_install_root/console/was/java/bin/keytool
 -storepasswd -new $NEW_PASSWORD$ -keystore $KEYSTORE_LOCATION$
 -storepass $CURRENT_PASSWORD$
 -storetype JKS
```

If the keystore 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:

```
/WBIC_install_root/console/was/bin
```

and execute the following command:

```
./wsadmin.sh -f /WBIC_install_root/console/scripts/
bcgChgPassword.jacl -conntype NONE
```

Repeat this step for the WebSphere Application Server instances of the Receiver and Document Manager.

You will be prompted for the new password.

Note: If a certificate in a truststore 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 truststore 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**.

Creating and installing certificates

The following sections describe how to create and install certificates that you want to use with WebSphere Business Integration Connect.

Inbound SSL certificates

If your community is not using SSL, neither you nor your participants need an inbound or outbound SSL certificate.

Server authentication

WebSphere Application Server uses the SSL certificate when it receives connection requests from participants through SSL. It is the certificate that the Receiver presents to identify the hub to the participant. 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 to generate a certificate and key pair. Refer to documentation available from IBM for more information about using ikeyman.

After you generate the certificate and key pair, use the certificate for inbound SSL traffic for all participants. If you have multiple Receivers or Consoles, copy the resultant keystore to each instance. If the certificate is self-signed, provide this certificate to the participants. To obtain this certificate, use ikeyman to extract the public certificate to a file.

If you are going to use self-signed server certificates, use one of the following procedures.

- **ikeyman**:
  1. Start the ikeyman utility, which is located in /WBIC_install_root/router/was/bin. If this is your first time using ikeyman, delete the "dummy" certificate that resides in the keystore.
  2. Use ikeyman to generate a self-signed certificate and a key pair for the Receiver or Console keystore.
  3. Use ikeyman to extract to a file the certificate that will contain your public key.
  4. Install the pkcs12 file into the Receiver or Console keystore for which it was created.
  5. Distribute the certificate to your participants. The preferred method for distribution is to send the certificate in a zip file that is password-protected, by e-mail. Your participants must call you and request the password for the zip file.
- **createCert.sh**:
  1. Use the createCert.sh script, located in the /WBIC_install_root/router/was/bin directory, to generate a self-signed certificate in X.509 format, a private key in PKCS 8 format, and a PKCS12 file which contains both the private key and certificate.
  2. Install the pkcs12 file into the Receiver or Console keystore for which it was created.
  3. Distribute the certificate to your participants. The preferred method for distribution is to send the certificate in a zip file that is password-protected, by e-mail. Your participants must call you and request the password for the zip file.

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 /WBIC_install_root/router/was/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 keystore.
5. Distribute the CA certificate to all participants.

Client authentication

For client authentication, use the following procedure:

1. Obtain your participant's certificate.
2. Install the certificate into the truststore using ikeyman.
3. Place the related CA in the CA directory or related keystore.

Note: When you add more participants to your hub community, you can use ikeyman to add their certificates to the truststore. If a participant leaves the community, you can use ikeyman to remove the participant's certificates from the truststore.

After installing the certificate, configure WebSphere Application Server to use client authentication by running the utility script bcgClientAuth.jacl.

- Navigate to the following directory: /WBIC_install_root/receiver/was/bin
- To turn on client authentication, call the script as follows: ./wsadmin.sh -f /WBIC_install_root/receiver/scripts/bcgClientAuth.jacl -conntype NONE set
- To turn off client authentication, call the script as follows: ./wsadmin.sh -f /WBIC_install_root/receiver/scripts/bcgClientAuth.jacl -conntype NONE clear

You must start the WebSphere Application Server receiver for these changes to take effect.

There is an additional feature that can be used with SSL client authentication. This feature is enabled via the Community Console. For HTTPS, WebSphere Business Integration Connect checks certificates against the Business IDs in the inbound documents. To use this feature, create the participant's profile, import the client certificate, and flag it as SSL. Select the **Validate Client SSL Certificate** option on the participant's Gateway screen.

Outbound SSL certificate

If your community is not using SSL, you do not need an inbound or outbound SSL certificate.

Server authentication

When SSL is being used to send outbound documents to your participants, WebSphere Business Integration Connect requests a server-side certificate from the participants. If a participant's certificate is self-signed, use the Community Console to import it into the Hub Operator profile and flag it as a **Root** certificate. If the certificate is CA-signed, you need only import the CA certificate into the Community Console and flag it as a **Root** certificate.

Note: The same CA certificate can be used for multiple participants. The certificate must be in X.509 DER format.

Client authentication

If SSL client authentication is required, the participant will, in turn, request a certificate from the hub. Use the Community Console to import your certificate into WebSphere Business Integration Connect. You can generate the certificate using ikeyman or the `createCert.sh` script. If the certificate is a self-signed certificate, it must be provided to the participant. If it is a CA-signed certificate, the CA root certificate must be given to the participants, so that they can add it to their trusted certificates.

If you are going to use a self-signed certificate, use one of the following procedures.

- **ikeyman**:
  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 participants. The preferred method for distribution is to send the certificate in a zip file that is password-protected, by e-mail. Your participants must call you and request the password for the zip 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. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Make sure you are logged in to the Community Console as the Hub Operator. Install the certificate in your own profile and flag it as an **SSL** type certificate.

- **createCert.sh**:
  1. Use the `createCert.sh` script to generate a self-signed certificate in X.509 format, a private key in PKCS 8 format, and a PKCS12 file which contains both the private key and certificate.
  2. Install the self-signed certificate and key through the Community Console. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Make sure you are logged in to the Community Console as the Hub Operator. Install the certificate in your own profile and flag it as an **SSL** type certificate.
  3. Send your self-signed certificate or CA root certificate to all participants so they can add it as a trusted certificate.

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 keystore.
4. Distribute the signing CA certificate to all participants.

Adding a Certificate Revocation List (CRL)

Business Integration Connect includes a Certificate Revocation List (CRL) feature. The CRL, issued by a Certificate Authority (CA), identifies participants who have revoked certificates before their scheduled expiration date. Participants with revoked certificates will be denied access to Business Integration Connect.

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.

CRLs are stored in the following location: `/<shared data directory>/security/crl`. Business Integration Connect uses the setting `bcg.http.CRLDir` in the `bcg.properties` file to identify the location of the CRL directory.

Create a `.crl` file containing the revoked certificates and place it in the CRL directory.

For example, in the `bcg.properties` file, you would use the following setting:

```
bcg.http.CRLDir=/<shared data directory>/security/crl.
```

Inbound signature certificate

The Document Manager uses the participant's signed certificate to verify the sender's signature when you receive documents. The participants send their self-signed signature certificates in X.509 DER format to you. You, in turn, install the participants' certificates through the Community Console under the respective participant's profile.

To install the certificate, use the following procedure.

1. Receive the participant's signature certificate in X.509 DER format.
2. Install the certificates through the Community Console under the participant's profile. Use **Account Admin > Profiles > Community Participant**, and search for the participant's profile. Click **Certificates**, and then upload the certificate as a **Digital Signature** certificate type. Do not forget to enable and save this certificate on the confirmation screen.
3. If the certificate was signed by a CA and the CA root certificate is not already installed in the Hub Operator profile, install it now. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Make sure you are logged in to the Community Console as the Hub Operator, and install the certificate in your own profile.

   Note: You do not have to perform the previous step if the CA certificate is already installed.
4. Enable at the package (highest level), participant, 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 participant connection, click **Account Admin > Participant Connections** and then select the participants. Click **Attributes** and then edit the attribute (for example, **AS Signed**).

Outbound signature certificate

The Document Manager uses this certificate when it sends outbound, signed documents to participants. The same certificate and key are used for all ports and protocols.

If you are going to use a self-signed certificate, use one of the following procedures.

ikeyman:

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 participants. The preferred method for distribution is to send the certificate in a zip file that is password protected, by e-mail. Your participants must call you and request the password for the zip 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 private key pair in the form of a PKCS12 file through the Community Console's certificate feature. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Make sure you are logged in to the Community Console as the Hub Operator, and install the

certificate in your own profile. Flag the certificate as type **Digital Signature**. Make sure you enable and save the certificate on the confirmation screen.

createCert.sh:

1. Use the createCert.sh script to generate a self-signed certificate in X.509 format, a private key in PKCS 8 format, and a PKCS12 file which contains both the private key and certificate.
2. Install the self-signed certificate and key through the Community Console's Certificates feature. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Make sure you are logged in to the Community Console as the Hub Operator, and install the certificate in your own profile. Flag the certificate as type **Digital Signature**. Make sure you enable and save the certificate on the confirmation screen.
3. Distribute the certificate to your participants. The preferred method for distribution is to send the certificate in a zip file that is password protected, by e-mail. Your participants must call you and request the password for the zip file.
4. Enable at the package (highest level), participant, 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 participant connection, click **Account Admin > Participant Connections** and then select the participants. Click **Attributes** and then edit the attribute (for example, **AS Signed**).

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 keystore.
5. Distribute the signing CA certificate to all participants.

Inbound encryption certificate

This certificate is used by the Receiver to decrypt encrypted files received from participants. The Receiver 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.

If you are going to use a self-signed certificate, use one of the following procedures.

- **ikeyman**:
  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 participants. 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 Community Manager. If your certificate is CA-signed, provide the CA certificate as well.
  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 private key pair in the form of a PKCS12 file through the Community Console. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Make sure you are logged in to the Community Console as the Hub Operator, and install the certificate in your own profile. Flag the certificate as an **Encryption** type and make sure you enable and save the installed certificate on the confirmation screen.
7. Enable at package (highest level), participant, 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 participant connection, click **Account Admin > Participant Connections** and then select the participants. Click **Attributes** and then edit the attribute (for example, **AS Encrypted**).

- **createCert.sh**:
  1. Use the createCert.sh script to generate a self-signed certificate in X.509 format, a private key in PKCS 8 format, and a PKCS12 file which contains both the private key and certificate.
  2. Install the self-signed certificate and key through the Console's certificate feature. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Make sure you are logged in to the Community Console as the Hub Operator, and install the certificate in your own profile. Flag the certificate as an **Encryption** type. Make sure you enable and save the installed certificate on the confirmation screen.
  3. Distribute the certificate to your participants. 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 Community Manager.
  4. Enable at package (highest level), participant, 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 participant connection, click **Account Admin > Participant Connections** and then select the participants. Click **Attributes** and then edit the attribute (for example, **AS Encrypted**).

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 keystore.
5. Distribute the signing CA certificate to all participants.

Outbound encryption certificate

The outbound encryption certificate is used when the hub sends encrypted documents to participants. Business Integration Connect encrypts documents with the public keys of the participants, and the participants decrypt the documents with their private keys.

1. Obtain the participant's encryption certificate. The certificate must be in X.509 DER format.
2. Install the certificate through the Community Console's Certificates feature. You perform this task logged in to the console as the Hub Operator, and install the certificate in the participant's profile. Use **Account Admin > Profiles > Community Participant**, and search for the participant's profile. Then click

Certificates and upload the certificate as an **Encryption** type certificate. Make sure you enable and save this certificate on the confirmation screen.

3. If the certificate is signed by a CA, and you do not have the CA's certificate installed in the system, log in to the console as Hub Operator and install this certificate in your own profile. Use **Account Admin > Profiles > Certificates** to display the Certificates page. Install the certificate in your own profile. You need only load a CA's certificate once.
4. Enable at package (highest level), participant, 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 participant connection, click **Account Admin > Participant Connections** and then select the participants. Click **Attributes** and then edit the attribute (for example, **AS Encrypted**).

Configuring Inbound SSL for the Console and Receiver

The WebSphere Business Integration Connect keystores are preconfigured in WebSphere Application Server. This section applies only if you are using different keystores.

To configure SSL for the Console and Receiver in Business Integration Connect, 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:

     ```
     WBIC_install_root/common/security/keystore/receiver.jks
     and
     WBIC_install_root/common/security/keystore/receiverTrust.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, JCEK, 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

   ```
   /WBIC_install_root/receiver/was/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.

   ```
   ./wsadmin.sh -f /WBIC_install_root/receiver/
   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 clientAuthentication with the value false.

Note: Repeat these steps for the Console, substituting **console** for **receiver** in the path name.

Chapter 8. Finishing the configuration

This chapter describes additional tasks you can perform to configure the hub.

Enabling the use of APIs

WebSphere Business Integration Connect 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 *Programmer Guide*.

Use this procedure to enable the use of the APIs so that participants can make API calls to the WebSphere Business Integration Connect server:

1. From the main menu, click **System Administration > Feature Administration > Administration API**.
2. Click the  next to **Enable the Administration API**.
3. Select the check box to enable the use of the API.
4. Click **Save**.

Specifying the queues used for events

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.

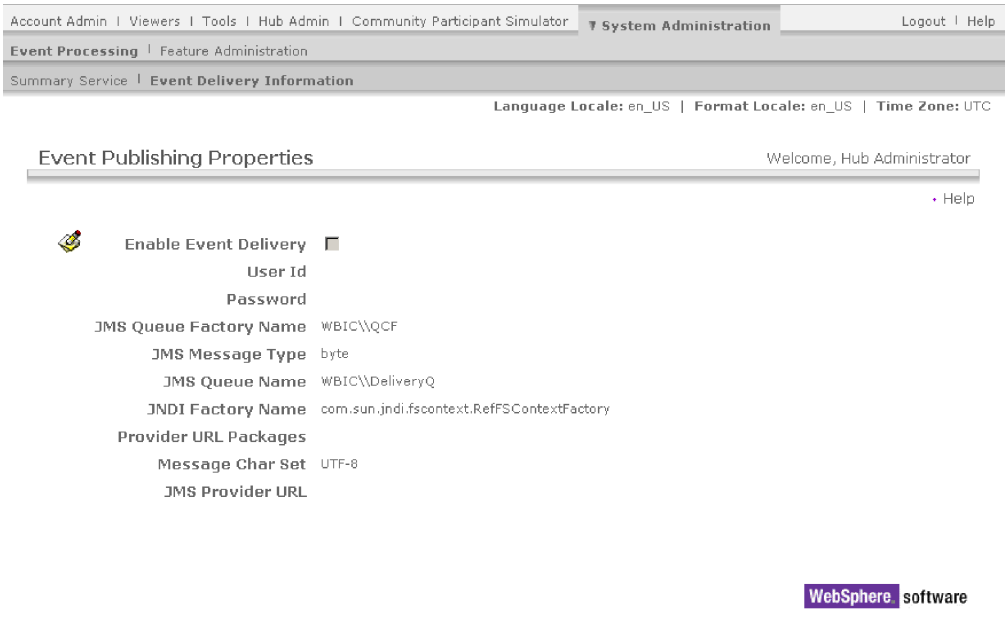


Figure 31. The Event Publishing Properties page

If you do not provide a value in the **Provider URL Packages** or the **JMS Provider URL** fields, the defaults that are in the MQ Properties section of the file <router-root-dir>/was/wbic/config/bcg.properties are used. These defaults use

the JMS bindings that were generated at installation time. If you took the defaults, the JMS bindings use port 9999 on the MQ Server that you named during installation.

To point to a different set of JMS bindings, change the **Provider URL Packages** to point to a directory containing a JMS bindings file that you have prepared yourself. Also change the **Queue Connection Factory** name and the **Queue name** to match the names you chose in your JMS bindings. You would do this if you want to publish the events to a queue on a different MQ server than the one you specified during installation.

To indicate where events should be delivered:

1. From the main menu, click **System Administration > Event Processing > Event Delivery Information**.
2. Click  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.
   - For **JMS Message Type**, enter the type of message that will be delivered. The choices are byte or text.
   - 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.
   - 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, which is <router-root-dir>/was/jndi/WBIC.
   - 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

When an event occurs within WebSphere Business Integration Connect, an event code is generated. Using the Event Codes screen, 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 screen. You can then set an alert for the event.

To indicate which events should be alertable:

1. **Click Hub Admin > Hub Configuration > Event Codes**.
   The Event Codes screen is displayed.
2. For each event you want made alertable:
   a. Click the magnifying glass icon next to the event code. The Event Code Details screen is displayed.
   b. Select **Alertable**.

Updating a user-defined transport

As described in Chapter 5, "Configuring the hub" and Chapter 6, "Creating participants and participant connections," you can upload an XML file that describes a user-defined transport. You use **Import Transport Type** to upload the file. After you upload the XML file, the transport becomes available for use when defining a target or gateway.

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 target or gateway page when you specify a user-defined transport. For example, a user-defined transport for a gateway might include the attribute GatewayRetryCount.

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 **Import Transport Type** to upload the file. Any change to the attributes are reflected in the gateway or target screen.

Appendix A. Examples

This appendix provides a basic example of setting up a hub, creating a participant and connections, and applying security for incoming and outgoing documents. It follows the order presented earlier in this book. Following the basic configuration example, you will find examples of configuring other transports and protocols.

Basic Configuration – Exchanging EDI documents with AS packaging over HTTP

In this example, the hub configuration is quite simple—two targets are defined (one for documents coming into the hub from a participant and one for documents coming into the hub from the Community Manager back-end system). The exchanges that are set up in this example use the Document Flow Definitions provided by WebSphere Business Integration Connect; therefore, you only have to create connections based on those flows. No custom XML is used in this example.

Configuring the hub

The first step in setting up the hub is creating the two targets.

- An HTTP Target (called "HttpTarget") to receive documents over HTTP (from Partner Two) that are to be sent to the back-end system of the Community Manager (Partner One)
- A File Directory Target (called "FileSystemTarget") to retrieve documents from the file system (from Partner One's back-end system) that are to be sent to Partner Two)

Defining the targets

To create a target for the receipt of HTTP:

1. Click **Hub Admin > Hub Configuration >Targets**.
2. Click **Create Target**.
3. For Target Name, type: **HttpTarget**.
4. From the Transport list, select **HTTP/S**.
5. For the Gateway type, use the default of **Production**.
6. For the URI, type: **/bcgreceiver/submit**
7. Click **Save**.

Next, you create a target to poll a directory on the file system. Creating the target automatically creates a new directory on the file system.

To create the file-system target:

1. Click **Hub Admin > Hub Configuration > Targets**.
2. Click **Create Target**.
3. For Target Name, type: **FileSystemTarget**.
4. From the Transport list, select **File Directory**.
5. For Default Gateway Type, use the default of **Production**.
6. For the Document Root Path, type: **\temp\FileSystemTarget**

Note: This will create a FileSystemTarget directory within the C:\temp directory. Be sure a C:\temp directory exists on the file system.

7. Click **Save**.

Defining document flows and interactions

In this example, you are setting up the following exchanges:

- Sending an EDI-X12 document, packaged in AS2, from Partner Two to Partner One
- Sending an EDI-X12 document, with no packaging, from Partner Two to Partner One.
- Sending an EDI-X12 document, packaged in AS2, from Partner One to Partner Two.
- Sending an EDI-X12 document, with no packaging, from Partner One to Partner Two

Because of the packaging and protocols involved, there is no need to create a new document flow definition. The packages, protocols, and document flows are ones that are predefined in the system.

However, you do need to define interactions based on these predefined document flows. You need two interactions:

- An interaction in which the source is an EDI-X12 document with no packaging and the target is an EDI-X12 document with AS2 packaging.
- An interaction in which the source is an EDI-X12 document in AS2 packaging and the target is an EDI-X12 document with no packaging.

Create the first interaction, in which the source format is an EDI-X12 document with no packaging and the target format is an EDI-X12 document with AS packaging.

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click **Manage Interactions** and then **Create Interaction**.
3. From the **Source** column select:
   a. Package: **None**
   b. Protocol: **EDI-X12**
   c. Document Flow: **All**
4. From the **Target** column select:
   a. Package: **AS**
   b. Protocol: **EDI-X12**
   c. Document Flow: **All**
5. Set **Action** to **Pass Through**.
6. Click **Save**.

Create a second interaction, in which the source format is an EDI-X12 document in AS packaging, and the target format is EDI-X12 with no packaging:

1. Click **Create Interaction**.
2. From the **Source** column, select:
   a. Package: **AS**
   b. Protocol: **EDI-X12**
   c. Document Flow: **All**
3. From the **Target** column, select:

a. Package: **None**
b. Protocol: **EDI-X12**
c. Document Flow: **All**

4. Set **Action** to **Pass Through**.
5. Click **Save**.

Creating participants and participant connections

In this example, one external participant is created, in addition to the Community Manager. The gateways for the participants include standard transports, and no configuration points are defined for the gateways.

Creating the participants

Create two new participants. To define Partner One:

1. Click **Account Admin** from the main menu. The Participant Search page is the default view.
2. Click **Create**.
3. For **Participant Login Name**, type: **partnerOne**
4. For **Participant Name**, type: **Partner One**
5. For **Participant Type**, select **Community Manager**.
6. Click **New** under **Business ID**.
7. Leave **Type** as **DUNS** and enter an Identifier value of **123456789**.
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 > Community Participant**.
2. Click **Create**.
3. For **Participant Login Name**, type: **partnerTwo**
4. For **Participant Name**, type: **Partner Two**
5. For **Participant Type**, select **Community Participant.**
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 Partner One and Partner Two to the hub.

The next steps are to configure gateways for both Partner One and Partner Two.

Creating the gateways

Before creating a file-directory gateway for Partner One, you must create the directory structure used by this gateway. Create a new FileSystemGateway directory on the root drive. This directory will be used by Partner One to store files received from participants.

In the case of Partner One, who is the Community Manager, the gateway represents the entrance point into the back-end system.

To create a gateway for Partner One:

1. Click **Account Admin > Profiles > Community Participant**.
2. Click **Search**.
3. Select **Partner One** by clicking the  icon.
4. Click **Gateways** from the horizontal navigation bar.
5. Click **Create**.
6. For **Gateway Name**, type: **FileSystemGateway**
7. For **Transport**, select **File Directory**.
8. For **Target URI file**, type: **file://C:\FileSystemGateway**
9. Click **Save**.

Next, set this newly created gateway as the default gateway for Partner One.

1. Click **List** to view all gateways configured for Partner One.
2. Click **View Default Gateways**.
3. From the list, select **FileSystemGateway** for the **Production Gateway Type**.
4. Click **Save**.

Create a gateway for Partner Two

1. Click **Account Admin > Profiles > Community Participant**.
2. Click **Search**, and then select **Partner Two** by clicking the magnifying glass icon.
3. Click **Gateways** from the horizontal navigation bar.
4. Click **Create**.
5. For **Gateway Name**, type: **HttpGateway**
6. For **Transport**, select **HTTP/1.1**.
7. For **Target URI file**, type: **http://<IP_address>:80/input/AS2**, where *<IP_address>* represents Partner Two's computer.
8. For **User Name**, type: **partnerOne**
9. For **Password**, type: **partnerOne**
10. Click **Save**.

Note that this example assumes that Parter Two requires a user name and password for any participant logging in to its system.

Again, you need to define a default gateway for this participant.

1. Click **List** followed by **View Default Gateways**.
2. From the list, select **HttpGateway** for the **Production Gateway Type**.
3. Click **Save**.

Setting up B2B Capabilities

Next, define the B2B Capabilities for Partner One (the Community Manager).

1. From the main menu, click **Account Admin > Profiles > Community Participant**.
2. Click **Search**.
3. Select **Partner One** by clicking the magnifying glass 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 Flow: ALL by performing the following steps:

a. Click the activate icon under **Set Source** for **Package: AS**
b. Click the activate icon under **Set Target** for **Package: AS**
c. Click the folder icon next to **Package: AS** to expand the folder.
d. Click the activate icon for **Protocol: EDI-X12 (ALL)** for both source and target.
e. Click the folder icon next to **Protocol: EDI-X12 (ALL)** to expand the folder
f. Click the activate icon for **Document Flow: ALL** for both source and target.

6. Set the Source and Target for Package: None, Protocol: EDI-X12, and Document Flow: ALL by performing the following steps:
   a. Click the activate icon under **Set Source** for **Package: None**
   b. Click the activate icon under **Set Target** for **Package: None**
   c. Click the folder icon next to **Package: None** to expand the folder.
   d. Click the activate icon for **Protocol: EDI-X12 (ALL)** for both source and target.
   e. Click the folder icon next to **Protocol: EDI-X12 (ALL)** to expand the folder
   f. Click the activate icon for **Document Flow: ALL** for both source and target.

Then, set the B2B Capabilities for Partner Two.

1. From the main menu, click **Account Admin > Profiles > Community Participant**.
2. Click **Search**.
3. Select Partner Two by clicking the magnifying glass 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 Flow: ALL by performing the following steps:
   a. Click the activate icon under **Set Source** for **Package: AS**
   b. Click the activate icon under **Set Target** for **Package: AS**
   c. Click the folder icon next to **Package: AS** to expand the folder.
   d. Click the activate icon for **Protocol: EDI-X12 (ALL)** for both source and target.
   e. Click the folder icon next to **Protocol: EDI-X12 (ALL)** to expand the folder
   f. Click the activate icon for **Document Flow: ALL** for both source and target.
6. Set the Source and Target for Package: None, Protocol: EDI-X12, and Document Flow: ALL by performing the following steps:
   a. Click the activate icon under **Set Source** for **Package: None**
   b. Click the activate icon under **Set Target** for **Package: None**
   c. Click the folder icon next to **Package: None** to expand the folder.
   d. Click the activate icon for **Protocol: EDI-X12 (ALL)** for both source and target.
   e. Click the folder icon next to **Protocol: EDI-X12 (ALL)** to expand the folder.
   f. Click the activate icon for **Document Flow: ALL** for both source and target.

Defining participant connections

Define the participant connection for EDI documents with no packaging that come from Partner One to be delivered to Partner Two.

1. Click **Account Admin > Participant Connections** .
2. From the **Source** list, select **Partner One**.
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 Flow: **ALL (ALL)**
   b. **Target**
      1) Package: **AS (N/A)**
      2) Protocol: **EDI-X12 (ALL)**
      3) Document Flow: **ALL (ALL)**

Next, define the connection for EDI documents wrapped in AS2 packaging that come from Partner Two to be delivered to Partner One 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 > Participant Connections**.
2. From the **Source** list, select **Partner Two**
3. From the **Target** list, select **Partner One**.
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 Flow: **ALL (ALL)**
   b. **Target**
      1) Package: **None (N/A)**
      2) Protocol: **EDI-X12 (ALL)**
      3) Document Flow: **ALL (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 screen and clicking the folder 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

In this section, you use the ikeyman tool 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 C:\ProgramFiles\IBM\WBIConnect\receiver\bin
2. Open the Receiver's default keystore, receiver.jks. From the menu bar, select **Key Database File Open**. On a default installation, receiver.jks resides in the directory:

   \WBIConnect\common\security\keystore
3. When prompted, enter the default password for receiver.jks. This password is WebAS.
4. Assuming this is the first time you have opened receiver.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 reference the public certificate (which was created as part of the creation of the self-signed certificate in the previous step) 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 tool.
2. Click **Extract Certificate**.
3. Change the Data type to **Binary DER data**.
4. Provide the file name **partnerOnePublic** 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 **partnerOnePrivate** 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 > Community Participant**.
2. Click **Search**.
3. Select Hub Operator by selecting the magnifying glass icon.
4. Click **Certificates** and then **Load Certificate**.
5. Set the **Certificate Type** as **Root 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 gateway to use secure HTTP.

1. Click **Account Admin > Profiles > Community Participant** from the horizontal navigation bar.
2. Click **Search** and select Partner Two by clicking the magnifying glass icon.
3. Click **Gateways** from the horizontal navigation bar. Next select HttpGateway by clicking the magnifying glass icon.
4. Edit it by clicking the edit icon.

5. Change the transport value to **HTTPS/1.1**
6. Change the value of the target URI 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

This section provides the steps for setting up encryption.

Partner Two must perform any necessary configuration steps (for example, importing the public certificate that was extracted from the self-signed certificate) and set up encryption on documents sent to the hub.

WebSphere Business Integration Connect 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 > Community Participant** from the horizontal navigation bar.
2. Click **Search**.
3. Select **Hub Operator** by clicking the magnifying glass icon.
4. Click **Certificates** and then click **Load PKCS12**.
5. Select the check box to the left of **Encryption**.
6. Change the Description to **Partner One Private**.
7. Select **Enabled**.
8. Click **Browse** and navigate to the directory in which the PKCS12 file, partnerOnePrivate.p12, is stored.
9. Select the file and click **Open**.
10. Enter the password provided for the PKCS12 file.
11. Leave the Gateway Type as **Production**.
12. Click **Upload**, and then click **Save**.

This completes the configuration required to allow a participant 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 keypair (in this example, partnerTwoDecrypt.der) and make it available to the hub.

As mentioned earlier, the public key will be used by the hub when encrypting transactions to be sent to the participant. In order to do so, you load the public certificate into the hub.

1. From the main menu, click **Account Admin > Profiles > Community Participan**t.
2. Click **Search**.
3. Select Partner Two by clicking the magnifying glass icon.

4. Click **Certificates** from the horizontal navigation bar.
5. Click **Load Certificate**.
6. Select the check box beside **Encryption**.
7. Change the Description to read **Partner Two Decrypt**.
8. Set the status as **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 Gateway Type as **Production**, click **Upload** and click **Save**.

The final step in configuring the hub to send encrypted messages over secure HTTP using AS2 is to modify the participant connection that exists between Partner One and Partner Two.

To modify the participant connection from the Community Console:

1. Click **Account Admin > Profiles > Participant Connections** from the horizontal navigation bar.
2. From the **Source list**, select **Partner One**.
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 folder icon next to **Package: AS (N\A)**.

   Note: You will need to scroll down the screen for this option to appear.
7. From the list, update the AS Encrypted attribute to **Yes** and click **Save**.

Setting up document signing

When digitally signing a transaction or message, WebSphere Business Integration Connect uses a participant's private key to create the signature and sign. Your partner receiving that message uses your public key to validate the signature. WebSphere Business Integration Connect uses Digital Signatures to this effect.

This section provides the steps required to configure both the hub and a participant 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 > Community Participant** from the horizontal navigation bar.
2. Click **Search**.
3. Select Partner Two by clicking the magnifying glass ion.
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 **Partner One 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 participant uses the public certificate imported as a Certifying Authority to authenticate signed transactions sent the hub.

The hub will use the private key to digitally sign outbound transactions sent to the participant. 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 magnifying glass icon next to **Hub Operator**.
3. Click the magnifying glass icon next to **Partner One Private**.

   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**.
6. Click **Save**.

Next you alter the attributes of the existing participant connection between Partner One and Partner Two to accommodate signed AS2.

To alter the attributes of the participant connection:

1. Click **Account Admin > Profiles > Participant Connections** from the horizontal navigation bar.
2. Select **Partner One** 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 folder 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 Business Integration Connect to the participant.

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 (a Community Manager named PartnerOne, using a DUNS ID of 123456789 and a file-directory gateway, and a participant named PartnerTwo with a DUNS ID of 987654321 and an HTTP gateway), this section describes how to add support for:

- The FTP transport
- Custom XML documents
- Binary files (with no packaging)

Creating an FTP target

The FTP target receives files and passes them to the Document Manager for processing. As described in Chapter 2, "Preparing to configure the hub," before you can create an FTP target, 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 > Targets**.
2. Click **Create**.
3. Enter the following information:
   a. Target 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 Partner One.

Creating a valid interaction for binary documents

By default, WebSphere Business Integration Connect does not have a binary-to-binary document interaction configured. In this section, you will create the required interaction to allow binary documents to pass through the system.

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click **Manage Interactions**.
3. Click **Create Interaction**.
4. From **Source** select: **Package: None Protocol: Binary (1.0) Document Flow: Binary (1.0)**.
5. From **Target** select: **Package: None Protocol: Binary (1.0) Document Flow: Binary (1.0)**.
6. From **Action** select **Pass Through**.
7. Click **Save**.

Updating the B2B capabilities for Partner One

This section shows how to configure Partner One to be able to accept binary documents.

1. Click **Account Admin > Profiles > Community Participant**.
2. Click **Search**.

3. Click the magnifying glass icon next to Partner One.
4. Click **B2B Capabilities**.
5. Click the activate icon underneath **Set Source** for **Package: None** to enable it.
6. Click the activate icon underneath **Set Target** for **Package: None** to enable it.
7. Click the folder icon next to **Package: None**.
8. Click the activate icon for **Protocol: Binary (1.0)** for both source and target.
9. Click the folder icon next to **Protocol: Binary (1.0)**.
10. Finally, click the activate icon for **Document Flow: Binary (1.0)** for both source and target.

Updating the B2B capabilities for Partner Two

This section shows how to configure Partner Two to be able to send binary documents.

1. Click **Account Admin > Profiles > Community Participant**.
2. Click **Search**.
3. Click the magnifying glass icon next to Partner Two.
4. Click **B2B Capabilities**.
5. Click the activate icon underneath **Set Source** for **Package: None** to enable.
6. Click the activate icon underneath **Set Target** for **Package: None** to enable.
7. Click the folder icon next to **Package: None**.
8. Click the activate icon for **Protocol: Binary (1.0)** for both source and target.
9. Click the folder icon next to **Protocol: Binary (1.0)**.
10. Finally, click the activate icon for **Document Flow: Binary (1.0)** for both source and target.

Creating a new participant connection

This section shows how to configure a new participant connection between Partner One and Partner Two for binary documents.

1. Click **Account Admin > Participant Connections**.
2. Select **Partner Two** from the **Source** list.
3. Select **Partner One** 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 Chapter 5, “Configuring the hub,” you must configure the hub to be able to route 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>
        <From>987654321</From>
       <To>123456789</To>
   </Tester>
```

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 Participant Name and To Participant Name.

Creating the CustomXML protocol definition format

The first step is to create a new protocol for the Custom XML you are going to exchange.

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click **Create Document Flow Definition**.
3. Enter the following information:
   a. Document flow type: **Protocol**
   b. Code: **CustomXML**
   c. Version: **1.0**
   d. Description: **CustomXML**
4. Set **Document Level** to **No**.
5. Set **Status** to **Enabled**.
6. Set **Visibility: Community Operator** to **Yes**.
7. Set **Visibility: Community Manager** to **Yes**.
8. Set **Visibility: Community Participant** to **Yes**.
9. Select:
   a. Package: **AS**
   b. Package: **None**
   c. Package: **Backend Integration**.
10. Click **Save**.

Creating the Tester_XML document definition

The second step is to create a document flow definition for the new protocol.

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click **Create Document Flow Definition**.
3. Enter the following information:
   a. Document flow type: **Document Flow**
   b. Code: **XML_Tester**
   c. Version: **1.0**
   d. Description: **XML_Tester**
4. Set **Document Level** to **Yes**.
5. Set **Status** to **Enabled**.
6. Set **Visibility: Community Operator** to **Yes**.
7. Set **Visibility: Community Manager** to **Yes**.
8. Set **Visibility: Community Participant** to **Yes**.
9. Click the folder icon next to Package: AS and select **Protocol: CustomXML**.
10. Click the folder icon next to Package: None and select **Protocol: CustomXML**.
11. Click the folder icon next to Package: Backend Integration and select **Protocol: CustomXML**.
12. Click **Save**.

Creating the Tester_XML XML Format

Finally, you create the XML format associated with the new protocol.

1. Click **Hub Admin > Hub Configuration > XML Formats**.
2. Click **Create XML Format**.
3. Enter the following information:
   a. Routing Format: **CustomXML 1.0**

b. File Type: **XML**
c. Identifier Type: **Root Tag**
d. Identifier Type Value: **Tester**
e. Source Business Id: **Element Path**
f. Source Business Id Value: **/Tester/From**
g. Target Business Id: **Element Path**
h. Target Business Id Value: **Tester/To**
i. Source Document Flow: **Constant**
j. Source Document Flow Value: **XML_Tester**
k. Source Document Flow Version: **Constant**
l. Source Document Flow Version Value: **1.0**

4. Click **Save**.

Creating a valid interaction for XML_Tester XML documents

You now have a new protocol and document flow with which to set up a valid interaction.

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click **Manage Interactions**.
3. Click **Create Interaction**.
4. From **Source**, select:
   a. Package: **None**
   b. Protocol: **CustomXML (1.0)**
   c. Document Flow: **XML_Tester (1.0)**.
5. From **Target** select:
   a. Package: **None**
   b. Protocol: **CustomXML (1.0)**
   c. Document Flow: **XML_Tester (1.0)**.
6. From **Action**, select **Pass Through**.
7. Click **Save**.

Updating the B2B capabilities for partnerOne

To enable the exchange of the custom XML document, you must update the B2B capabilities of the participants.

1. Click **Account Admin > Profiles > Community Participant**.
2. Click **Search**.
3. Click the magnifying glass icon next to Partner One.
4. Click **B2B Capabilities**.
5. Click the activate icon underneath **Set Source** for Package: None to enable it.
6. Click the activate icon underneath **Set Target** for Package: None to enable it.
7. Click the folder icon next to Package: None.
8. Click the activate icon for Protocol: CustomXML (1.0) for both source and target.
9. Click the folder icon next to Protocol: CustomXML (1.0).
10. Finally, click the activate icon for Document Flow: XML_Tester (1.0) for both source and target.

Updating the B2B capabilities for partnerTwo

You update the B2B capabilities of Partner Two to enable the exchange of the new custom XML format.

1. Click **Account Admin > Profiles> Community Participant**.
2. Click **Search**.
3. Click the magnifying glass icon next to Partner Two.
4. Click **B2B Capabilities**.
5. Click the activate icon underneath **Set Source** for Package: None to enable it.
6. Click the activate icon underneath **Set Target** for Package: None to enable it.
7. Click the folder icon next to Package: None.
8. Click the activate icon for Protocol: CustomXML (1.0) for both source and target.
9. Click the folder icon next to Protocol: CustomXML (1.0).
10. Finally, click the activate icon for Document Flow: XML_Tester (1.0) for both source and target.

Creating a new participant connection

Finally, create a new participant connection.

1. Click **Account Admin > Participant Connections.**
2. Select **Partner Two** from the **Source** list.
3. Select **Partner One** 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.

Appendix B. Setting up RosettaNet exchanges

RosettaNet is an organization that provides open standards to support the exchange of business messages between trading partners. For more information on RosettaNet, see http://www.rosettanet.org. The standards include RosettaNet Implementation Framework (RNIF) and Partner Interface Process (PIP) specifications. RNIF defines how trading 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 Business Integration Connect 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 Business Integration Connect provides a protocol for packaging the transformed message into a RosettaNet Service Content (RNSC) message that the back-end system can handle. See the Enterprise Integration Guide for information on 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 participant). You provide the Document Flow Definitions for the RNIF version and the PIPs you want to use.

In addition to providing routing capability for RosettaNet messages, WebSphere Business Integration Connect 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 participants if it receives certain Event Notification messages from back-end systems. See the Enterprise Integration Guide for more information on Event Notification.

RNIF and PIP document flow packages

To support RosettaNet messaging, WebSphere Business Integration Connect provides two sets of ZIP files called packages. The *RNIF packages* consist of Document Flow Definitions required to support the RNIF protocol. These packages are in the B2BIntegrate directory.

For RNIF V1.1

- Package_RNIF_1.1.zip
- Package_RNSC_1.0_RNIF_1.1.zip

For RNIF V02.00

- Package_RNIF_V02.00.zip
- Package_RNSC_1.0_RNIF_V02.00.zip

The first package in each pair provides the Document Flow Definitions required to support RosettaNet communications with participants, and the second package provides the Document Flow Definitions required to support RosettaNet communications with back-end systems.

The second set of packages consists of PIP document flow packages. Each PIP document flow package has a Packages directory containing an XML file and a GuidelineMaps directory containing XSD files. The XML file specifies the Document Flow Definitions that define how WebSphere Business Integration Connect handles the PIP and define the exchanged messages and signals. The XSD files specify the format of the PIP's messages and define acceptable values for XML elements in the messages. The ZIP 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 Business Integration Connect provides PIP document flow packages are:

- PIP 0A1 Notification of Failure v1.0
- PIP 0A1 Notification of Failure V02.00.00
- PIP 2A12 Distribute Product Master V01.03.00
- PIP 3A1 Request Quote V02.00.00
- PIP 3A2 Request Price and Availability R02.01.00B
- 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 Order Update V02.02.00
- PIP 3A8 Request Purchase Order Change V01.02.00
- PIP 3A9 Request Purchase Order Cancellation V01.01.00
- PIP 3B12 Shipping Order Request V01.01
- PIP 3B13 Shipping Order Confirmation Notification V01.01
- PIP 3B18 Shipping Documentation Notification V01.00
- PIP 3B2 Notify of Advance Shipment V01.01.00
- 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 Self Billing Invoice Notification V01.00
- 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 Planning Release Forecast Notification V02.00
- PIP 4A5 Notify of Forecast Reply V02.00.00
- PIP 4B2 Notify of Shipment Receipt V01.00.00
- PIP 4C1 Distribute Inventory Report V02.03.00
- PIP 4C1 Inventory Report Notification V02.01
- PIP 5C1 Distribute Product List V01.00.00
- PIP 5C4 Distribute Registration Status V01.02.00
- PIP 5D1 Request Ship From Stock And Debit Authorization V01.00.00
- PIP 7B1 Distribute Work in Process V01.00.00
- PIP 7B5 Notify of Manufacturing Work Order V01.00
- PIP 7B6 Notify of Manufacturing Work OrderReply V01.00

For each PIP, there are four PIP document flow packages:

- For RNIF 1.1 messaging with participants
- For RNIF 1.1 messaging with back-end systems
- For RNIF 2.0 messaging with participants
- For RNIF 2.0 messaging with back-end systems

Each PIP document flow package follows a specific naming convention so that you can identify whether the package is for messages between WebSphere Business Integration Connect and participants or between WebSphere Business Integration Connect and back-end systems. The naming convention also identifies the RNIF version, PIP, and PIP version that the package supports. For PIP document flow packages used for messaging between WebSphere Business Integration Connect and participants, the format is:

BCG_Package_RNIF<*RNIF version*>_<*PIP*><*PIP version*>.zip

For PIP document flow packages used for messaging between WebSphere Business Integration Connect 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 participants and WebSphere Business Integration Connect using the RNIF 1.1 protocol. For PIP document flow 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 Enterprise Integration Guide for information on the packaging used for these messages.

Setting up RosettaNet support

For RosettaNet messaging, WebSphere Business Integration Connect requires the RNIF packages for the version of RNIF used to send the messages. For each PIP that Business Integration Connect supports, it requires the PIP's two PIP document flow packages for the RNIF version. For example, to support the 3A4 PIP over RNIF 2.0, Business Integration Connect 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 participants and the second package supports RosettaNet messaging with back-end systems. The third package and fourth packages enable Business Integration Connect to pass 3A4 messages between participants and back-end systems using RNIF 2.0.

To support RosettaNet messaging:

1. If Business Integration Connect does not have the RNIF packages loaded for the version of RNIF you want to use, import them. See "Uploading RNIF packages" on page 38 for information on how to import the packages into Business Integration Connect.
2. For each PIP you want to support, upload the PIP document flow package for the PIP and for the RNIF version you are supporting. For information on the convention used to name these packages, see "RNIF and PIP document flow

packages" on page 91. If Business Integration Connect 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 flow packages" on page 99 for more information.

Creating connections to participants

The following process describes how to create a connection between a back-end system and a participant. Note that you must create a connection for each PIP that you want to send and one for each PIP that you want to receive.

Before you begin, ensure that the following conditions apply:

- You are logged in as a Hub Admin.
- The appropriate RNIF Document Flow Definitions have been uploaded and that the packages for the PIP you want to use have been uploaded. See "Setting up RosettaNet support" on page 93 for the names of these packages.

To create a connection for a particular PIP, do the following:

1. Create the interaction for the connection:
   a. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
   b. Click **Manage Interactions**.
   c. Click **Create Interaction**.
   d. Expand the source Document Flow Definition tree to the Action level and expand the target Document Flow Definition tree to the Action level.
   e. In the trees, select the Document Flow Definitions to use for the source context and the target context. For example, if the participant is the initiator of a 3C6 PIP (a one-action PIP), select the following Document Flow Definitions in the trees:

Table 1. 3C6 PIP initiated by a participant

Source	Target
Package: RNIF (V02.00)	Package: Backend Integration (1.0)
Protocol: RosettaNet (V02.00)	Protocol: RNSC (1.0)
Document Flow: 3C6 (V01.00)	Document Flow: 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 Flow Definitions from the trees:

Table 2. 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 Flow: 3C6 (V01.00)	Document Flow: 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 participant, select the following Document Flow Definitions for the first action:

Table 3. 3A4 PIP initiated by a participant

Source	Target
Package: RNIF (V02.00)	Package: Backend Integration (1.0)
Protocol: RosettaNet (V02.00)	Protocol: RNSC (1.0)
Document Flow: 3A4 (V02.02)	Document Flow: 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 Flow Definitions for the first action:

Table 4. 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 Flow: 3A4 (V02.02)	Document Flow: 3A4 (V02.02)
Activity: Request Purchase Order	Activity: Request Purchase Order
Action: Purchase Order Request Action	Action: Purchase Order Request Action

f. In the Action field, select **Bi-Directional Translation of RosettaNet and RosettaNet Service Content with Validation**.

g. Click **Save**.

h. If you are setting up a two-action PIP, repeat steps c-g to create the interaction for the second action. For example, select the following Document Flow Definitions for the second action for a 3A4 PIP initiated by a participant. This is the action in which the back-end system sends the response.

Table 5. 3A4 PIP initiated by a participant (second action)

Source	Target
Package: Backend Integration (1.0)	Package: RNIF (V02.00)
Protocol: RNSC (1.0)	Protocol: RosettaNet (V02.00)
Document Flow: 3A4 (V02.02)	Document Flow: 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 Flow Definitions:

Table 6. 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 Flow: 3A4 (V02.02)	Document Flow: 3A4 (V02.02)
Activity: Request Purchase Order	Activity: Request Purchase Order
Action: Purchase Order Confirmation Action	Action: Purchase Order Confirmation Action

2. If a participant profile does not exist for the participant, create it. See "Creating participants" on page 49 for information on how to do this. There must also be a participant profile of the Community Manager type for the back-end system.
3. If a gateway with the supported protocol does not exist between the participant and Business Integration Connect or between a back-end system and Business Integration Connect, create it. See "Creating gateways" on page 50 for information on how to do this. The supported protocols for RosettaNet messages between a participant and Business Integration Connect are HTTP

and HTTPS. The supported protocols for RosettaNet messages between a back-end system and Business Integration Connect are HTTP, HTTPS, and JMS.

4. Activate the Document Flow Definitions that Business Integration Connect uses to process the PIP. To do this, activate the participant's and back-end system's definitions for the Package, Protocol, and Document Flow for the PIP. The direction of the message determines which one is the source and which one is the target. Business Integration Connect automatically activates the Activity, Actions, and Signals when you activate the parent Document Flow. For information on how to activate the Document Flow Definitions, see "Setting up B2B capabilities" on page 57.

   Participant

   - Package: RNIF (1.1 or V02.00 depending on which RNIF version you are using)
   - Protocol: RosettaNet (1.1 or V02.00 depending on which RNIF version you are using)
   - Document Flows: <*PIP name and version*>

   back-end system

   - Package: Backend Integration (1.0)
   - Protocol: RNSC (1.0)
   - Document Flows: <*PIP name and version*>

5. Activate the connection by setting the source and target in the Participant Connections screen. If the participant is the initiator of the PIP, set the source to the participant's profile and the target to the Community Manager profile. If the initiator is a back-end system, set the source to the Community Manager profile and set the target to the participant's profile. See "Creating interactions" on page 46 for information on searching for connections and activating them. If the PIP is a two-action PIP, you must also activate the connection in the other direction 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.
6. If Business Integration Connect does not have a target defined for each of the protocols, create it. See "Setting up targets" on page 30 for information on how to do this.

Editing RosettaNet attribute values

For RosettaNet support, an Action type Document Flow 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 Business Integration Connect automatically define values for these attributes and you usually do not need to change them.

To edit the RosettaNet attributes of an Action Document Flow Definition, do the following:

1. Click **Hub Admin > Hub Configuration > Document Flow Definition.**
2. Click folder icons to individually expand a node to the appropriate Document Flow Definition level or select All to expand the entire tree.
3. The Actions column for each Action Document Flow Definition contains a RosettaNet attributes icon. Click this icon to edit the RosettaNet attributes of the Action. The 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 7. 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 participant or back-end system that is sending the message
To Service	Contains the network component service name of the participant or back-end system that is receiving the message
From Role	Contains the role name of the participant or back-end system that is sending the message
To Role	Contains the role name of the participant or back-end system that is receiving the message
Root Tag	Contains the name of the root element in the PIP message's XML document
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 functionality of the Document Flow Definition.
6. Click **Save**.

Configuring attribute values

For PIP Document Flow 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 participant. 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 participant 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 Business Integration Connect provides PIP document flow 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 participant.

If you want to set the attributes using the Document Flow Definition context, do the following.

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Click folder icons to individually expand a node to the appropriate Document Flow Definition level or select **All** to expand all displayed Document Flow Definition nodes.
3. In the **Actions** column, click the Edit attribute values icon for the package you want to edit such as Package:RNIF (1.1) or Package:RNIF (V02.00).
4. In the **Document Flow Context Attributes** section, go to the **Update** column of the attribute you want to set and select or type the new value in the update field. Repeat for each attribute that you want to set.
5. Click **Save**.

If you want to set the value of the attributes for each connection, do the following:.

1. Click **Account Admin > Participant Connections**.
2. Select the source and the target of the connection you want to change and then click **Search**.
3. The Console displays a list of connections that match the source and target criteria. Each connection displays two sets of Document Flow Definitions (Source and Target) and a set of buttons including two **Attributes** buttons. To edit Document Flow Definition attributes for the source or target, click the **Attributes** button closest to the source or target you want to edit.
4. In the Connection Attributes window, expand the Package node.
5. Go to the **Update** column of the attribute you want to set and select or type the new value in the update field. Repeat for each attribute that you want to set.
6. Click **Save**.

Deactivating PIPs

After a PIP package has been uploaded into Business Integration Connect, it cannot be removed. However, you can deactivate the PIP so that it cannot be used.

To deactivate a PIP for all communications with participants, do the following:

1. Click **Hub Admin > Hub Configuration > Document Flow Definition**.
2. Expand the Document Flow Definitions tree to reveal the Document Flow Definition of the PIP you want to disable.

3. In the Status column of the package, click **Enabled**. The Status column now displays ″Disabled″ and Business Integration Connect cannot use the Document Flow Definition for the PIP.

To deactivate a PIP communication with a specific participant, deactivate the connection to the participant defined for the PIP.

Providing failure notification

If a failure occurs during the processing of a PIP message, Business Integration Connect uses the 0A1 PIP as the mechanism to broadcast the failure to the participant or back-end system that sent the message. For example, say a back-end system initiates a 3A4 PIP. Business Integration Connect processes the RNSC message and sends a RosettaNet message to a participant. Business Integration Connect waits for the response to the RosettaNet message until the waiting time reaches the timeout limit. Once this occurs, Business Integration Connect creates a 0A1 PIP and sends it to the participant. The 0A1 PIP identifies the exception condition so that the participant 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 participant 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 <install_root>/wbic/config directory.

These fields populate the contact information within the 0A1 PIP. Fax is optional (value can be empty), but the rest are required.

- **bcg.0A1.fromContactName**
- **bcg.0A1.fromEMailAddr**
- **bcg.0A1.fromPhoneNbr**
- **bcg.0A1.fromFaxNbr**

The phone numbers are limited to 30 bytes in length. The other fields are unlimited in length. When they are changed, the router will need to be restarted.

Creating PIP document flow packages

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 flow package for PIP 5C4 V01.03.00. Business Integration Connect supplies a PIP document flow package for PIP 5C4 V01.02.00 so the procedures actually document how to perform an upgrade. However, creating a PIP document flow 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 flow package involves the following procedures:

- Creating the XSD files
- Creating the XML file
- Creating the packages

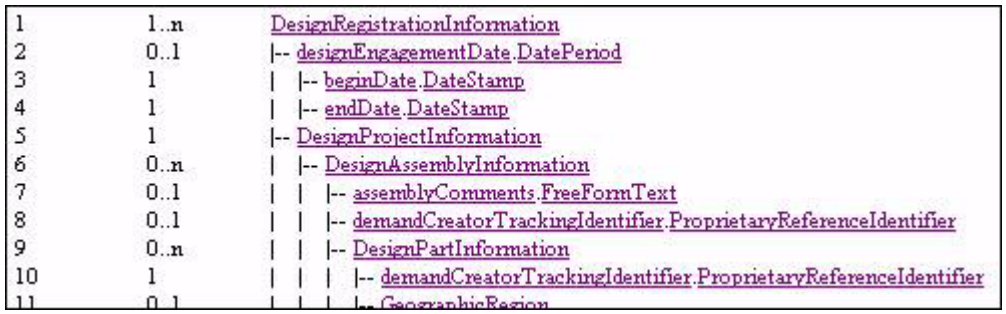
Creating the XSD files

A PIP document flow 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 on the XSD files, see "About validation" on page 109.

To create the XSD files for the PIP document flow package, do the following:

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. 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 flow package based of another version of the package and want to reuse a definition from the other version, do the following for each of these definitions:

a. Delete the definition of the element. For example, delete the ContactInformation element.

b. Open the PIP document flow 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 flow 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:

```
name="ContactInformation
type="ContactInformation_type7"
```

If you are creating a PIP document flow package, or are upgrading a PIP document flow package but the definition you need does not exist in the other version, do the following 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:

```
name="ContactInformation
type="ContactInformation_localType1"
```

   Element productProviderFieldApplicationEngineer before modification

```
<xsd:element name="productProviderFieldApplicationEngineer">
  <xsd:complexType>
    <xsd:sequence>
```

```
      <xsd:element ref="ContactInformation"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Element productProviderFieldApplicationEngineer after modification

```
<xsd:element name="productProviderFieldApplicationEngineer">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="ContactInformation"
          type="ContactInformation_localType1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

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, GlobalRegistrationComplexityLevelCode has the following table:

GlobalRegistrationComplexityLevelCode	lines 139
Entity Instances	
Above average	Above average complexity
Average	Average complexity
Maximum	Maximum complexity
Minimum	Minimal complexity
None	No complexity
Some	Some complexity

Therefore, in a PIP 5C4 V01.03.00 message, the GlobalRegistrationComplexityLegelCode can only have the following values: Above average, Average, Maximum, Minimum, None and Some.

If you are updating the PIP document flow package based on another version of the package and want to reuse a set of enumeration values from the other version, do the following for each set:

a. Delete the definition for the element. For example, delete the GlobalRegistrationComplexityLevelCode element:

b. Open the PIP document flow 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 flow 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:

```
name="GlobalRegistrationComplexityLevelCode"
type="_GlobalRegistrationComplexityLevelCode"
```

If you are creating a PIP document flow package or are upgrading a PIP document flow package but the enumeration value definitions you need do not exist in the other version, do the following for any element with enumerated values in the guidelines:

a. Delete the definition of the element. For example, delete the GlobalRegistrationComplexityLegelCode element.

b. Create the replacement definition. For example, create the GlobalRegistrationComplexityLegelCode_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:

```
name="GlobalRegistrationComplexityLevelCode"
type="GlobalRegistrationComplexityLevelCode_localType"
```

Element DesignAssemblyInformation before modification

```
<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>
```

Element DesignAssemblyInformation after modification

```
<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>
```

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 as shown in the following figure:

Fundamental Business Data Entities

Name	Definition	Data Type	Min	Max	Representation
CommunicationsNumber	The electro-technical communication number, e.g., telephone number, facsimile number, pager number.	String	1	30	X(30)
DateStamp	Specifies a specific date. Date stamp based on the ISO 8601 specification. The "Z" following the day identifier (DD) is used to indicate Coordinated Universal Time. Informal format: YYYYMMDDZ	Date	9	9	9(8)X

If you are updating the PIP document flow package based on another version of the package and want to reuse a data entity definition from the other version, do the following for each set:

a. Delete the definition for the data entity element. For example, delete the DateStamp element:

b. Open the PIP document flow 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 flow 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:

```
name="DateStamp" type="_common_DateStamp_R"
```

If you are creating a PIP document flow package or are upgrading a PIP document flow package but the data entity definition you need does not exist in the other version, do the following 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:

```
name="DateStamp" type="DateStamp_localType"
```

Element beginDate before modification

```
<xsd:element name="beginDate">
  <xsd:complexType">
    <xsd:sequence>
      <xsd:element ref="DateStamp"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Element beginDate after modification

```
<xsd:element name="beginDate">
  <xsd:complexType">
    <xsd:sequence>
      <xsd:element name="DateStamp" type="DateStamp_localType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Creating the XML file

After you have created the XSD files for your PIP document flow 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_RNIFV02.00_5C4V01.03.zip and BCG_RNSC1.0_RNIFV02.00_5C4V01.03.zip respectively. The following procedure describes how to create the XML file for the RNIF package:

1. Extract the XML file from a RNIF PIP document flow package file. If you are upgrading, extract the file from the previous version of the package such as BCG_Package_RNIFV02.00_5C4V01.02.zip. If you are creating a new package, extract the file from a PIP document flow 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 such as 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 may appear more than once in the file, so make sure that you update all instances.

Table 8. 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_5C4RegistrationStatusNotification_ V01.02	BCG_5C4RegistrationStatusNotification_ V01.03
The name of the confirmation message XSD file without the file extension (for two-action PIPs only)	N/A	N/A
Root element name in the XSD file for the request message	Pip5C4RegistrationStatusNotification	Pip5C4RegistrationStatusNotification
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 may not have to update these values.

Table 9. 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 may not have to update these values.

Table 10. 5C4 PIP attribute updates

Information to update	Description	Value in the 5C4 package	Element path in the XML file

Table 10. 5C4 PIP attribute updates (continued)

NonRepudiationRequired	Specified in Table 3-3	N	ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is NonRepudiationRequired) ns1:AttributeValue AttributePickListItem ATTRVALUEKEY
NonRepudiationOfReceipt	Specified in Table 3-3	N	ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is NonRepudiationOfReceipt) ns1:AttributeValue AttributePickListItem ATTRVALUEKEY
DigitalSignatureRequired	Specified in Table 5-1	Y	ns1:Package ns1:Protocol ns1:Process ns1:Attribute (Its ATTRIBUTEKEY is DigitalSignatureRequired) ns1:AttributeValue AttributePickListItem ATTRVALUEKEY
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 as shown in the following example for BCG_common.xsd.

To create the Backend Integration package, repeat the above procedure except for the following differences:

- In step 1, extract the XML file from the Backend Integration package such as BCG_Package_RNSC1.0_RNIFV02.00_5C4V01.02.zip.
- Do not do step 5.

After you have created the XML and the XSD files, you are ready to create the PIP documentation flow packages.

Creating the package

To create the RNIF package, do the following:

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 flow 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 the above procedure 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 Uploading RNIF packages.

About validation

Business Integration Connect 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 flow package, Business Integration Connect 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 flow packages" on page 99 for information on the steps needed to upgrade the XSD files used to validate the messages and how to create a custom PIP document flow 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 Business Integration Connect 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″ respectively 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 _GlobalLocationIdentifier 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 value.

PIP document flow package contents

The following table shows the PIP document flow packages provided by Business Integration Connect 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 flow packages for the PIP.

Table 11. PIP document flow package contents

Package ZIP file name	Packages contents	GuidelineMaps contents
PIP 2A12 Distribute Product Master		
BCG_Package_ RNIF1.1_2A12V01.03.zip	BCG_RNIF1.1_2A12V01.03.xml	BCG_2A12ProductMaster Notification_V01.03.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd
BCG_Package_ RNIFV02.00_2A12V01.03.zip	BCG_RNIFV02.00_ 2A12V01.03.xml	BCG_GlobalAssemblyLevelCode.xsd BCG_GlobalIntervalCode.xsd BCG_GlobalLeadTimeClassification Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd
BCG_Package_RNSC1.0_ RNIF1.1_2A12V01.03.zip	BCG_RNSC1.0_ RNIF1.1_2A12V01.03.xml	BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalProductLifeCycleStatus Code.xsd BCG_GlobalProductProcurementType Code.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_2A12V01.03.zip	BCG_RNSC1.0_ RNIFV02.00_2A12V01.03.xml	BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 3A1 Request Quote		
BCG_Package_ RNIF1.1_3A1V02.00.zip	BCG_RNIF1.1_3A1V02.00.xml	BCG_3A1QuoteConfirmation_V02.00.xsd BCG_3A1QuoteRequest_V02.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalCurrencyCode.xsd
BCG_Package_ RNIFV02.00_3A1V02.00.zip	BCG_RNIFV02.00_3A1V02.00.xml	BCG_GlobalGovernmentPriorityRating Code.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3A1V02.00.zip	BCG_RNSC1.0_ RNIF1.1_3A1V02.00.xml	BCG_GlobalQuoteTypeCode.xsd BCG_GlobalStockIndicatorCode.xsd BCG_GlobalTaxExemptionCode.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalProductSubstitutionReason Code.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_3A1V02.00.zip	BCG_RNSC1.0_ RNIFV02.00_3A1V02.00.xml	BCG_GlobalProductTermsCode.xsd BCG_GlobalQuoteLineItemStatusCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 3A2 Request Price and Availability		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A2R02.01.zip	BCG_RNIF1.1_3A2R02.01.xml	BCG_3A2PriceAndAvailabilityRequest_ R02.01.xsd BCG_3A2PriceAndAvailabilityResponse_ R02.01.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd
BCG_Package_ RNIFV02.00_3A2R02.01.zip	BCG_RNIFV02.00_3A2R02.01.xml	BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalPricingTypeCode.xsd BCG_GlobalProductStatusCode.xsd BCG_GlobalCurrencyCode.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3A2R02.01.zip	BCG_RNSC1.0_ RNIF1.1_3A2R02.01.xml	BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalCustomerAuthorization Code.xsd BCG_GlobalProductAvailabilityCode.xsd BCG_GlobalProductSubstitutionReason
BCG_Package_RNSC1.0_ RNIFV02.00_3A2R02.01.zip	BCG_RNSC1.0_ RNIFV02.00_3A2R02.01.xml	Code.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 3A4 Request Purchase Order		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A4V02.02.zip	BCG_RNIF1.1_3A4V02.02.xml	BCG_3A4PurchaseOrder Confirmation_V02.02.xsd BCG_3A4PurchaseOrder Request_V02.02.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalAccountClassificationCode.xsd BCG_GlobalCreditCardClassification Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalFinanceTermsCode.xsd BCG_GlobalGovernmentPriorityRating Code.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPaymentConditionCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalPriceUnitOfMeasureCode.xsd BCG_GlobalPurchaseOrderFillPriority Code.xsd BCG_GlobalPurchaseOrderTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalSpecialFulfillmentRequest Code.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_InvoiceChargeTypeCode.xsd BCG_GlobalTaxExemptionCode.xsd BCG_GlobalConfirmationTypeCode.xsd BCG_GlobalProductSubstitutionReason Code.xsd BCG_GlobalPurchaseOrder AcknowledgmentReasonCode.xsd BCG_GlobalPurchaseOrderStatusCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 3A4PurchaseOrderRequest		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A4V02.00.zip	BCG_RNIF1.1_3A4V02.00.xml	BCG_3A4PurchaseOrder Request_V02.00.xsd BCG_3A4PurchaseOrder Confirmation_V02.00.xsd BCG_common_V422.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PhysicalAddress_Types_V422.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalAccountClassificationCode.xsd BCG_GlobalCreditCardClassification Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalDocumentReferenceType Code_V422.xsd BCG_GlobalFinanceTermsCode.xsd BCG_GlobalGovernmentPriorityRating Code.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalPartnerRoleClassification Code_V422.xsd BCG_GlobalPaymentConditionCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalPriceUnitOfMeasureCode.xsd BCG_GlobalPurchaseOrderFillPriority Code.xsd BCG_GlobalPurchaseOrderTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd BCG_GlobalShipmentTermsCode_V422.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalSpecialFulfillmentRequest Code.xsd BCG_InvoiceChargeTypeCode.xsd BCG_GlobalTaxExemptionCode.xsd BCG_GlobalTaxExemptionCode_V422.xsd BCG_GlobalSpecialHandling Code_V422.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessDescription_Types_V422.xsd BCG_GlobalConfirmationTypeCode.xsd BCG_GlobalProductSubstitutionReason Code.xsd BCG_GlobalPurchaseOrder AcknowledgmentReasonCode.xsd BCG_GlobalPurchaseOrderStatusCode.xsd BCG_common.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 3A5 Query Order Status		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A5R02.00.zip	BCG_RNIF1.1_3A5R02.00.xml	BCG_3A5PurchaseOrderStatus Query_R02.00.xsd BCG_3A5PurchaseOrderStatus Response_R02.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCreditCardClassification Code.xsd BCG_GlobalAccountClassification Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalFinanceTermsCode.xsd BCG_GlobalGovernmentPriorityRating Code.xsd BCG_GlobalPurchaseOrder AcknowledgmentReasonCode.xsd BCG_GlobalPurchaseOrderStatus Code.xsd BCG_GlobalShippingServiceLevel Code.xsd BCG_GlobalSpecialFulfillmentRequest Code.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_GlobalLineItemStatusCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalOrderQuantityTypeCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalProductSubstitutionReason Code.xsd BCG_GlobalTaxExemptionCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPurchaseOrderFillPriority Code.xsd BCG_GlobalPurchaseOrderTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd BCG_GlobalFreeOnBoardCode.xsd BCG_GlobalTransportEventCode.xsd BCG_GlobalCustomerTypeCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalCountryCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 3A6 Distribute Order Status		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A6V02.02.zip	BCG_RNIF1.1_3A6V02.02.xml	BCG_3A6PurchaseOrderStatus Notification_V02.02.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalAccountClassificationCode.xsd BCG_GlobalCreditCardClassification Code.xsd
BCG_Package_ RNIFV02.00_3A6V02.02.zip	BCG_RNIFV02.00_3A6V02.02.xml	BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalFinanceTermsCode.xsd BCG_GlobalGovernmentPriorityRating Code.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalLineItemStatusCode.xsd BCG_GlobalNotificationReasonCode.xsd BCG_GlobalOrderQuantityTypeCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3A6V02.02.zip	BCG_RNSC1.0_ RNIF1.1_3A6V02.02.xml	BCG_GlobalPaymentConditionCode.xsd BCG_GlobalPriceUnitOfMeasureCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalProductSubstitutionReason Code.xsd BCG_GlobalPurchaseOrderFillPriority Code.xsd BCG_GlobalPurchaseOrder AcknowledgmentReasonCode.xsd BCG_GlobalPurchaseOrderStatusCode.xsd BCG_GlobalPurchaseOrderTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_3A6V02.02.zip	BCG_RNSC1.0_ RNIFV02.00_3A6V02.02.xml	BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalSpecialFulfillmentRequest Code.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_GlobalTrackingReferenceType Code.xsd BCG_InvoiceChargeTypeCode.xsd BCG_GlobalTaxExemptionCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 3A7 Notify of Purchase Order Update		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A7V02.02.zip	BCG_RNIF1.1_3A7V02.02.xml	BCG_3A7PurchaseOrderUpdate Notification_V02.02.xsd BCG_common.xsd BCG_string_len_0.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalActionCode.xsd BCG_GlobalCountryCode.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalAccountClassificationCode.xsd BCG_GlobalConfirmationTypeCode.xsd BCG_GlobalCreditCardClassification Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalFinanceTermsCode.xsd BCG_GlobalGovernmentPriorityRating Code.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPaymentConditionCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalProductSubstitutionReason Code.xsd BCG_GlobalPriceUnitOfMeasureCode.xsd BCG_GlobalPurchaseOrderFillPriority Code.xsd BCG_GlobalPurchaseOrder AcknowledgmentReasonCode.xsd BCG_GlobalPurchaseOrderStatusCode.xsd BCG_GlobalPurchaseOrderTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalSpecialFulfillmentRequest Code.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_InvoiceChargeTypeCode.xsd BCG_GlobalTaxExemptionCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_xml.xsd
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	
PIP 3A8 Request Purchase Order Change		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A8V01.02.zip	BCG_RNIF1.1_3A8V01.02.xml	BCG_3A8PurchaseOrderChange Confirmation_V01.02.xsd BCG_3A8PurchaseOrderChange Request_V01.02.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalActionCode.xsd
BCG_Package_ RNIFV02.00_3A8V01.02.zip	BCG_RNIFV02.00_3A8V01.02.xml	BCG_GlobalAccountClassificationCode.xsd BCG_GlobalCreditCardClassification Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalFinanceTermsCode.xsd BCG_GlobalGovernmentPriorityRating Code.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3A8V01.02.zip	BCG_RNSC1.0_ RNIF1.1_3A8V01.02.xml	BCG_GlobalPaymentConditionCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalPriceUnitOfMeasureCode.xsd BCG_GlobalPurchaseOrderFillPriority Code.xsd BCG_GlobalPurchaseOrderTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalSpecialFulfillmentRequest Code.xsd BCG_GlobalSpecialHandlingCode.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_3A8V01.02.zip	BCG_RNSC1.0_ RNIFV02.00_3A8V01.02.xml	BCG_InvoiceChargeTypeCode.xsd BCG_GlobalTaxExemptionCode.xsd BCG_GlobalConfirmationTypeCode.xsd BCG_GlobalProductSubstitution ReasonCode.xsd BCG_GlobalPurchaseOrder AcknowledgmentReasonCode.xsd BCG_GlobalPurchaseOrderStatusCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 3A9 Request Purchase Order Cancellation		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3A9V01.01.zip	BCG_RNIF1.1_3A9V01.01.xml	BCG_3A9PurchaseOrderCancellation Confirmation_V01.01.xsd BCG_3A9PurchaseOrderCancellation Request_V01.01.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalPurchaseOrderCancellation Code.xsd BCG_GlobalPurchaseOrderCancellation ResponseCode.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
BCG_Package_ RNIFV02.00_3A9V01.01.zip	BCG_RNIFV02.00_3A9V01.01.xml	
BCG_Package_RNSC1.0_ RNIF1.1_33A9V01.01.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	
PIP 3B2 Notify of Advance Shipment		
BCG_Package_ RNIF1.1_3B2V01.01.zip	BCG_RNIF1.1_3B2V01.01.xml	BCG_3B2AdvanceShipment Notification_V01.01.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalIncotermsCode.xsd BCG_GlobalShipmentChangeDisposition Code.xsd BCG_GlobalShipmentModeCode.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalShipDateCode.xsd BCG_GlobalPackageTypeCode.xsd BCG_GlobalPhysicalUnitOfMeasure Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalLotQuantityClassification Code.xsd BCG_NationalExportControl ClassificationCode.xsd BCG_GlobalCountryCode.xsd BCG_GlobalTrackingReferenceType Code.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalPartnerClassification Code.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 3B12ShippingOrder Request		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3B12V01.01.zip	BCG_RNIF1.1_3B12V01.01.xml	BCG_3B12ShippingOrderRequest_ V01.01.xsd BCG_3B12ShippingOrderConfirmation_ V01.01.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_ContactInformation_Types_V422.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_PartnerDescription_Types_V422.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_GlobalIncotermsCode.xsd BCG_GlobalPackageTypeCode.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPhysicalUnitOfMeasure Code.xsd BCG_GlobalShipDateCode.xsd BCG_common_V422.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 3B13ShippingOrder ConfirmationNotification		
BCG_Package_ RNIF1.1_3B13V01.01.zip	BCG_RNIF1.1_3B13V01.01.xml	BCG_3B13ShippingOrderConfirmation Notification_V01.01.xsd BCG_common_V422.xsd BCG_ContactInformation_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalCurrencyCode.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalPhysicalUnitOfMeasure Code.xsd BCG_GlobalShipDateCode.xsd BCG_GlobalTrackingReferenceType Code.xsd BCG_common.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_PhysicalAddress_Types.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	

Table 11. PIP document flow package contents (continued)

PIP 3B18ShippingDocumentation Notification		
BCG_Package_ RNIF1.1_3B18V01.00.zip	BCG_RNIF1.1_3B18V01.00.xml	BCG_3B18ShippingDocumentation Notification_V01.00.xsd BCG_common_V422.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessDescription_Types_V422.xsd BCG_PhysicalAddress_Types.xsd BCG_ContactInformation_Types.xsd BCG_InvoiceChargeTypeCode_V422.xsd BCG_NationalExportControl ClassificationCode.xsd BCG_GlobalPartnerRoleClassification
BCG_Package_ RNIFV02.00_3B18V01.00.zip	BCG_RNIFV02.00_3B18V01.00.xml	Code_V422.xsd BCG_GlobalPartnerClassification Code_V422.xsd BCG_GlobalShippingDocument Code_V422.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalOrderAdminCode_V422.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPhysicalUnitOfMeasure Code_V422.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3B18V01.00.zip	BCG_RNSC1.0_ RNIF1.1_3B18V01.00.xml	BCG_GlobalCurrencyCode.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalIncotermsCode.xsd BCG_GlobalPaymentTermsCode.xsd BCG_GlobalTrackingReferenceType Code.xsd BCG_GlobalSpecialHandling Code_V422.xsd BCG_GlobalProductUnitOfMeasure Code.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_3B18V01.00.zip	BCG_RNSC1.0_ RNIFV02.00_3B18V01.00.xml	BCG_GlobalPackageTypeCode_V422.xsd BCG_GlobalPortTypeCode_V422.xsd BCG_GlobalPortIdentifierAuthority Code_V422.xsd BCG_GlobalShipDateCode.xsd BCG_GlobalFreeOnBoardCode_V422.xsd BCG_GlobalFreightPaymentTerms Code_V422.xsd BCG_GlobalShipmentModeCode.xsd BCG_GlobalShippingServiceLevel Code.xsdBCG_string_len_0.xsd
PIP 3C3 Notify of Invoice		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3C3V01.01.zip	BCG_RNIF1.1_3C3V01.01.xml	BCG_3C3InvoiceNotification_V01.01.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalDocumentTypeCode.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPaymentTermsCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalSaleTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_InvoiceChargeTypeCode.xsd BCG_NationalExportControl ClassificationCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 3C4 Notify of Invoice Reject		
BCG_Package_ RNIF1.1_3C4V01.00.zip	BCG_RNIF1.1_3C4V01.00.xml	BCG_3C4InvoiceReject Notification_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalInvoiceRejectionCode.xsd BCG_GlobalMonetaryAmountType Code.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 3C6 Notify of Remittance Advice		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3C6V01.00.zip	BCG_RNIF1.1_3C6V01.00.xml	BCG_3C6RemittanceAdvice Notification_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd
BCG_Package_ RNIFV02.00_3C6V01.00.zip	BCG_RNIFV02.00_3C6V01.00.xml	BCG_PartnerDescription_Types.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalFinancialAdjustment ReasonCode.xsd BCG_GlobalInvoiceRejectionCode.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3C6V01.00.zip	BCG_RNSC1.0_ RNIF1.1_3C6V01.00.xml	BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalPaymentMethodCode.xsd BCG_GlobalDocumentTypeCode.xsd BCG_BusinessDescription_Types.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_3C6V01.00.zip	BCG_RNSC1.0_ RNIFV02.00_3C6V01.00.xml	BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 3C7SelfBillingInvoice Notification		
BCG_Package_ RNIF1.1_3C7V01.00.zip	BCG_RNIF1.1_3C7V01.00.xml	BCG_3C7SelfBillingInvoice Notification_V01.00.xsd BCG_common_V422.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd
BCG_Package_ RNIFV02.00_3C7V01.00.zip	BCG_RNIFV02.00_3C7V01.00.xml	BCG_NationalExportControl ClassificationCode.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessDescription_Types_V422.xsd BCG_InvoiceChargeTypeCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalCurrencyCode.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3C7V01.00.zip	BCG_RNSC1.0_ RNIF1.1_3C7V01.00.xml	BCG_GlobalMonetaryAmountType Code.xsd BCG_GlobalDocumentTypeCode.xsd BCG_GlobalDocumentTypeCode_V422.xsd BCG_GlobalPaymentTermsCode.xsd BCG_GlobalSaleTypeCode.xsd BCG_GlobalShipmentTermsCode.xsd BCG_GlobalCountryCode.xsd
BCG_Package_RNSC1.0_ RNIF1.1_3C7V01.00.zip	BCG_RNSC1.0_ RNIFV02.00_3C7V01.00.xml	BCG_GlobalShippingServiceLevelCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_common.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 3D8 Distribute Work in Process		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_3D8V01.00.zip	BCG_RNIF1.1_3D8V01.00.xml	BCG_3D8WorkInProcess Notification_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalPriorityCode.xsd BCG_GlobalWorkInProcessLocation Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalWorkInProcessPartType Code.xsd BCG_GlobalLotCode.xsd BCG_GlobalLotStatusCode.xsd BCG_GlobalLotQuantityClassification Code.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalCountryCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 4A1 Notify of Strategic Forecast		
BCG_Package_ RNIF1.1_4A1V02.00.zip	BCG_RNIF1.1_4A1V02.00.xml	BCG_4A1StrategicForecast Notification_V02.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalForecastEventCode.xsd BCG_GlobalForecastTypeCode.xsd BCG_GlobalPartnerReferenceType Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_StrategicForecastQuantityType Code.xsd BCG_GlobalForecastIntervalCode.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 4A3 Notify of Threshold Release Forecast		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_4A3V02.00.zip	BCG_RNIF1.1_4A3V02.00.xml	BCG_4A3ThresholdRelease ForecastNotification_V02.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd
BCG_Package_ RNIFV02.00_4A3V02.00.zip	BCG_RNIFV02.00_4A3V02.00.xml	BCG_GlobalCurrencyCode.xsd BCG_GlobalForecastEventCode.xsd BCG_GlobalPartnerReferenceType Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd
BCG_Package_RNSC1.0_ RNIF1.1_4A3V02.00.zip	BCG_RNSC1.0_ RNIF1.1_4A3V02.00.xml	BCG_GlobalForecastIntervalCode.xsd BCG_GlobalForecastReferenceType Code.xsd BCG_GlobalForecastInventoryType Code.xsd BCG_OrderForecastQuantityTypeCode.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_4A3V02.00.zip	BCG_RNSC1.0_ RNIFV02.00_4A3V02.00.xml	BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 4A4PlanningRelease ForecastNotification		
BCG_Package_ RNIF1.1_4A4R02.00A.zip	BCG_RNIF1.1_4A4R02.00A.xml	BCG_4A4PlanningReleaseForecast Notification_R02.00A.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PhysicalAddress_Types_V422.xsd
BCG_Package_ RNIFV02.00_4A4R02.00A.zip	BCG_RNIFV02.00_4A4R02.00A.xml	BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalForecastReferenceType Code.xsd BCG_GlobalPartnerReference TypeCode.xsd
BCG_Package_RNSC1.0_ RNIF1.1_4A4R02.00A.zip	BCG_RNSC1.0_ RNIF1.1_4A4R02.00A.xml	BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalIntervalCode.xsd BCG_GlobalTransportEventCode.xsd BCG_GlobalForecastQuantityType Code_V422.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_4A4R02.00A.zip	BCG_RNSC1.0_ RNIFV02.00_4A4R02.00A.xml	BCG_GlobalCurrencyCode.xsd BCG_GlobalForecastInventoryType Code.xsd BCG_GlobalCountryCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 4A5 Notify of Forecast Reply		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_4A5V02.00.zip	BCG_RNIF1.1_4A5V02.00.xml	BCG_4A5ForecastReply Notification_V02.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalForecastEventCode.xsd BCG_GlobalPartnerReferenceType Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalForecastIntervalCode.xsd BCG_GlobalForecastReferenceType Code.xsd BCG_GlobalForecastResponseCode.xsd BCG_GlobalForecastInventoryType Code.xsd BCG_GlobalForecastRevisionReason Code.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_ForecastReplyQuantityTypeCode.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
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_4A5V02.00.xml	
BCG_Package_RNSC1.0_ RNIFV02.00_4A5V02.00.zip	BCG_RNSC1.0_ RNIFV02.00_4A5V02.00.xml	
PIP 4B2 Notify of Shipment Receipt		
BCG_Package_ RNIF1.1_4B2V01.00.zip	BCG_RNIF1.1_4B2V01.00.xml	BCG_4B2ShipmentReceipt Notification_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalLotDiscrepancyReason Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalReceivingDiscrepancyReason Code.xsd BCG_GlobalReceivingDiscrepancy Code.xsd BCG_GlobalSpecialFulfillmentRequest Code.xsd BCG_GlobalSpecialHandlingCode.xsd BCG_GlobalTrackingReferenceType Code.xsd BCG_BusinessDescription_Types.xsd BCG_GlobalPartnerClassification Code.xsd BCG_string_len_0.xsd BCG_xml.xsd
BCG_Package_ RNIFV02.00_4B2V01.00.zip	BCG_RNIFV02.00_4B2V01.00.xml	
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	

Table 11. PIP document flow package contents (continued)

PIP 4C1 Distribute Inventory Report		
BCG_Package_ RNIF1.1_4C1V02.03.zip	BCG_RNIF1.1_4C1V02.03.xml	BCG_4C1InventoryReport Notification_V02.03.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalInventoryCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalPartnerClassification Code.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 4C1Inventory ReportNotification		
BCG_Package_ RNIF1.1_4C1V02.01.zip	BCG_RNIF1.1_4C1V02.01.xml	BCG_4C1InventoryReport Notification_V02.01.xsd BCG_common_V422.xsd BCG_ContactInformation_Types.xsd BCG_ContactInformation_Types_V422.xsd BCG_PhysicalAddress_Types_V422.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalInventoryCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_common.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_PhysicalAddress_Types.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 5C1 Distribute Product List		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_5C1V01.00.zip	BCG_RNIF1.1_5C1V01.00.xml	BCG_5C1ProductList Notification_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd
BCG_Package_ RNIFV02.00_5C1V01.00.zip	BCG_RNIFV02.00_5C1V01.00.xml	BCG_PartnerDescription_Types.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPartnerClassificationCode.xsd
BCG_Package_RNSC1.0_ RNIF1.1_5C1V01.00.zip	BCG_RNSC1.0_ RNIF1.1_5C1V01.00.xml	BCG_GlobalCountryCode.xsd BCG_GlobalPriceTypeCode.xsd BCG_GlobalCurrencyCode.xsd BCG_BusinessDescription_Types.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_5C1V01.00.zip	BCG_RNSC1.0_ RNIFV02.00_5C1V01.00.xml	BCG_BusinessTaxIdentifier_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 5C4 Distribute Registration Status		
BCG_Package_ RNIF1.1_5C4V01.02.zip	BCG_RNIF1.1_5C4V01.02.xml	BCG_5C4RegistrationStatus Notification_V01.02.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd
BCG_Package_ RNIFV02.00_5C4V01.02.zip	BCG_RNIFV02.00_5C4V01.02.xml	BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalRegistrationComplexity LevelCode.xsd
BCG_Package_RNSC1.0_ RNIF1.1_5C4V01.023.zip	BCG_RNSC1.0_ RNIF1.1_5C4V01.02.xml	BCG_GlobalRegistrationInvolvement LevelCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPartnerClassificationCode.xsd
BCG_Package_RNSC1.0_ RNIFV02.00_5C4V01.02.zip	BCG_RNSC1.0_ RNIFV02.00_5C4V01.02.xml	BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_string_len_0.xsd BCG_xml.xsd
PIP 5D1 Request Ship From Stock And Debit Authorization Status		

Table 11. PIP document flow package contents (continued)

BCG_Package_ RNIF1.1_5D1V01.00.zip	BCG_RNIF1.1_5D1V01.00.xml	BCG_5D1ShipFromStockAnd DebitAuthorization Confirmation_V01.00.xsd BCG_5D1ShipFromStockAnd DebitAuthorizationRequest_V01.00.xsd BCG_common.xsd BCG_BusinessDescription_Types.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalCurrencyCode.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalPriceTypeCode.xsd BCG_GlobalShipFromStockAnd DebitAuthorizationRejectionCode.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 7B1 Distribute Work in Process		
BCG_Package_ RNIF1.1_7B1V01.00.zip	BCG_RNIF1.1_7B1V01.00.xml	BCG_7B1WorkInProcess Notification_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PhysicalAddress_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalChangeReasonCode.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalEquipmentTypeCode.xsd BCG_GlobalLotCode.xsd BCG_GlobalLotStatusCode.xsd BCG_GlobalLotQuantityClassification Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPriorityCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalWorkInProcessTypeCode.xsd BCG_GlobalWorkInProcessQuantity ChangeCode.xsd BCG_GlobalWorkInProcessLocation Code.xsd BCG_GlobalWorkInProcessPartType Code.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_string_len_0.xsd BCG_xml.xsd
BCG_Package_ RNIFV02.00_37B1V01.00.zip	BCG_RNIFV02.00_7B1V01.00.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	

Table 11. PIP document flow package contents (continued)

PIP 7B5NotifyOfManufacturing WorkOrder		
BCG_Package_ RNIF1.1_7B5V01.00.zip	BCG_RNIF1.1_7B5V01.00.xml	BCG_7B5NotifyOfManufacturing WorkOrder_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalBusinessActionCode_V422.xsd BCG_GlobalAttachmentDescription Code_V422.xsd BCG_GlobalMimeType QualifierCode_V422.xsd BCG_GlobalDevicePackageType Code_V422.xsd BCG_GlobalPackageTypeCode.xsd BCG_GlobalChangeReasonCode.xsd BCG_GlobalLineItemStatusCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalPhysicalUnitOfMeasure Code.xsd BCG_GlobalWorkInProcessLocation Code.xsd BCG_GlobalLotCode.xsd BCG_GlobalPriorityCode.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_PhysicalAddress_Types.xsd BCG_string_len_0.xsd BCG_xml.xsd
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	
PIP 7B6NotifyOfManufacturing WorkOrderReply		
BCG_Package_ RNIF1.1_7B6V01.00.zip	BCG_RNIF1.1_7B6V01.00.xml	BCG_7B6NotifyOfManufacturing WorkOrderReply_V01.00.xsd BCG_common.xsd BCG_ContactInformation_Types.xsd BCG_PartnerDescription_Types.xsd BCG_GlobalProductUnitOfMeasure Code.xsd BCG_GlobalDocumentReferenceType Code.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_GlobalChangeReasonCode.xsd BCG_GlobalLineItemStatusCode.xsd BCG_BusinessDescription_Types.xsd BCG_BusinessTaxIdentifier_Types.xsd BCG_GlobalCountryCode.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_PhysicalAddress_Types.xsd BCG_string_len_0.xsd BCG_xml.xsd
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_ RNIF1.1_7B6V01.00.zip	BCG_RNSC1.0_ RNIFV02.00_7B6V01.00.xml	

Table 11. PIP document flow package contents (continued)

PIP 0A1 Notification of Failure v1.0		
BCG_Package_ RNIF1.1_0A11.0.zip	BCG_RNIF1.1_0A11.0.xml	0A1FailureNotification_1.0.xml BCG_0A1FailureNotification_1.0.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_common.xsd BCG_string_len_0.xsd BCG_xml.xsd
BCG_Package_RNSC1.0_ RNIF1.1_0A11.0.zip	BCG_RNSC1.0_ RNIF1.1_0A11.0.xml	
PIP 0A1 Notification of Failure V02.00.00		
BCG_Package_ RNIF1.1_0A1V02.00.zip	BCG_RNIF1.1_0A1V02.00.xml	0A1FailureNotification_V02.00.xml BCG_0A1FailureNotification_V02.00.xsd BCG_common.xsd BCG_GlobalPartnerClassificationCode.xsd BCG_GlobalPartnerRoleClassification Code.xsd BCG_string_len_0.xsd BCG_xml.xsd
BCG_Package_ RNIFV02.00_0A1V02.00.zip	BCG_RNIFV02.00_0A1V02.00.xml	
BCG_Package_RNSC1.0_ RNIF1.1_0A1V02.00.zip	BCG_RNSC1.0_ RNIF1.1_0A1V02.00.xml	
BCG_Package_RNSC1.0_ RNIFV02.00_0A1V02.00.zip	BCG_RNSC1.0_ RNIFV02.00_0A1V02.00.xml	

Appendix C. Setting up Web service requests

A participant can request a Web service provided by the Community Manager. Similarly, the Community Manager can request a Web service provided by a participant. The participant or Community Manager invokes the WebSphere Business Integration Connect server to obtain the Web service. WebSphere Business Integration Connect acts as a proxy, passing the Web service request to the Web service provider and returning the response synchronously from the provider to the requestor.

This appendix contains the following information for setting up a Web service for use by a participant or a Community Manager:

- Identifying the participants for a Web service
- Setting up a Document Flow Definition for a Web service
- Adding Document Flow Definitions to participant B2B capabilities
- Activating the participant connection
- Restrictions and limitations of Web service support

Identifying the participants for a Web service

When a Web service is provided by the Community Manager for use by participants, WebSphere Business Integration Connect requires that a participant identify itself. When posting the Web service request, set the identity in one of the following two ways:

1. Use HTTP Basic Authentication with User ID of the form:
   - `<participant's business ID>/<console user name>` (for example, 123456789/joesmith).
   - Password equal to the console user name's password.
2. Present an SSL client certificate that has been previously loaded into WebSphere Business Integration Connect for the participant

When the Web service is provided by a participant, for use by the Community Manager, the public URL used by the Community Manager to invoke the Web service should contain the query string '?to=<participant's business ID>'. An example is:

```
http://WBIChost/bcgreceiver/Receiver?to=123456789
```

This tells WebSphere Business Integration Connect that the provider of the Web service is the participant with business ID '123456789'.

Setting up Document Flow Definitions for a Web service

To set up the Document Flow Definition, you upload the WSDL (Web Service Definition Language) files that define the Web service, as described in Chapter 5, "Configuring the hub." Alternatively, you can enter the equivalent Document Flow Definitions manually through the Community Console.

To enter the equivalent Document Flow Definitions manually, follow the procedures in "Creating a document definition flow" on page 43. You must also create the Document Flow, Activity and Action entries individually under the

Protocol Web Service, as described below, paying particular attention to the requirements for the Action and its relationship to the received SOAP messages.

In terms of the Package/Protocol/Document Flow/Activity/Action hierarchy of Document Flow Definitions, a supported Web service is represented as:

Package: None (name and code), version N/A
Protocol: Web Service (name and code), version 1.0
Document Flow: '{<web service namespace>}:<web service name>' (name and code), which is required to be unique among document flows for the Web Service protocol. This is typically the WSDL's namespace and name
Activities: One activity for each Web service operation, with name and code:

```
'{<operation namespace>}:<operation name>'
```

Actions: One action for the input message of each operation, with name and code:

```
'{<namespace of identifying xml element = first child of soap:body>}:<name of
identifying xml element = first child of soap:body>'
```

The critical definitions are the Actions because WebSphere Business Integration Connect 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 participant 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 Business Integration Connect's Document Flow Definitions.

For example, if a Web service request SOAP message is as follows (for a Document-Literal SOAP binding):

```
<?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>
```

Then WebSphere Business Integration Connect 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 Business Integration Connect 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.

Uploading the WSDL files for a Web service

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 may be uploaded with the primary file using one of the following methods:

- If the file path or (HTTP) URL in each import element's "location" attribute 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 zip file, each with a zip path corresponding to the path (if any) in the import "location" attribute, uploading the zip file will upload all the contained primary and imported WSDL files.

Example:

Primary WSDL file 'helloworldRPC.wsdl' contains

```
'<import namespace="http://www.helloworld.com/wsdl/helloRPC.wsdl" location=
"bindingRPC.wsdl"/>'
```

Imported WSDL file 'bindingRPC.wsdl' contains

```
'<import namespace="http://www.helloworld.com/wsdl/helloRPC.wsdl" location=
"port/porttypeRPC.wsdl"/>'
```

Zip 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 by WebSphere Business Integration Connect. They are passed through directly, with the original service end-point URL.) This is called the *private* WSDL.

In addition a *public* WSDL is saved with the private URL replaced by a target URL, as provided by the user in the Document Flow Upload input. 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 Business Integration Connect will then route the Web service request to a gateway that is the original Web service provider's private URL. WebSphere Business Integration Connect 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

Business Integration Connect 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 ZIP archive.

Important: The WSDL files within the ZIP archive must be within a directory specified in the WSDL import element. For example, with the following import element: <import namespace="*http://www.helloworld.com/wsdl/helloRPC.wsdl*" location="*path1/bindingRPC.wsdl*"/>, the directory structure within the ZIP archive would be *path1/bindingRPC.wsdl*. In the next 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 ZIP archive.

To upload a single WSDL file or ZIP archive, use the following procedure.

1. Click **Hub Admin** > **Hub Configuration** > **Document Flow Definition**.
2. Click **Upload/Download Packages**.
3. Select **Yes** for WSDL Package to upload a WSDL file. For **Web Service Public URL**, enter the public URL of the Web service provided by the Community Manager (which will be invoked by a participant). For example, http(s)://<*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 participant (which will be invoked by the Community Manager), enter the public URL of the participant with a query string. For example, http(s)://<*target host:port*>/bcgreceiver/Receiver?to=<participant business ID>.
4. Click **Browse** and select the WSDL file or ZIP archive.
5. 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.
6. For **Overwrite Data**, select **Yes** to replace a file currently in the database. Select **No** to add the file to the database.
7. 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 Business Integration Connect 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`

Setting up an interaction for a new Web service

The final step in creating the necessary Document Flow Definitions for a new Web service is to set up an interaction with the same Web service Document Flow Action as both the Source and the Target.

To create interactions, use the following procedure.

1. Click **Hub Admin** > **Hub Configuration** > **Document Flow Definition**.
2. Click **Manage Interactions**.
3. Click **Create Interaction**.
4. Select **Pass Through** from the **Action** dropdown at the bottom of the screen (**Pass Through** is the only valid option supported in WebSphere Business Integration Connect for a Web service).

Adding Document Flows to Participants B2B Capabilities

Add the Web service document flows to the source and target participants' B2B capabilities to set up a participant connection between the source and target participants.

Before you set up a participant connection between the Web service user and the Web service provider, you need to set up the gateways that will be used in the participant connection. See "Creating gateways" on page 50.

The source gateway's URL is not used by the Web service. It can be a dummy URL. The source gateway can be used to set the **Validate Client IP** or **Validate Client SSL Cert** options on the sender side.

For the target gateway, specify the private URL supplied by the Web service provider. This is where WebSphere Business Integration Connect will invoke the Web service when it acts as a proxy for the Web service provider.

Activating the participant connection

The new document flow should appear as an available choice for participant connections between the two selected participants. Activate the participant connection to make the Web service available to the Source participant. See "Activating participant connections" on page 58.

Restrictions and Limitations of Web service support

WebSphere Business Integration Connect 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:

- 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).
- Soap With Attachments is not supported.

Appendix D. Setting up cXML exchanges

This appendix contains an overview of cXML support and information on creating document flow definitions for cXML exchanges.

cXML support overview

The WebSphere Business Integration Connect 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, Business Integration Connect does not provide cXML DTDs. You can download them from www.cxml.org; and then upload them into Business Integration Connect through the Validation Map module in the Community Console. After you upload the DTD, associate it with the cXML document flow. Refer to Chapter 5, “Configuring the hub” for more information on associating the DTD with the cXML document flow.

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 example below shows the From and To elements as the source and destination of the cXML document:

```
<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.

Business Integration Connect 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 flow definition in Business Integration Connect. 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 flow definitions within Business Integration Connect:

cXML element	Document flow definition
cXML DOCTYPE	Protocol
DTD version	Protocol version
Request (type) For example, OrderRequest	Document flow

Response

The target participant sends a cXML response to inform the source participant 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 request document and document flow definitions within Business Integration Connect:

cXML element	Document flow definition
cXML DOCTYPE	Protocol
DTD version	Protocol version
Response (type) For example, ProfileResponse	Document flow

Message

A cXML message contains the Business Integration Connect document flow 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 flow created in Business Integration Connect. In the example below, SubscriptionChangeMessage would be the document flow:

```
<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 flow definitions within Business Integration Connect:

cXML element	Document flow definition
cXML DOCTYPE	Protocol
DTD version	Protocol version
Message	Document flow

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 - Indicates whether the message is a test document or a production document. Allowed values are production (default) or test.
- inReplyTo - Specifies to which message this message responds. The contents of the inReplyTo attribute would be 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 listed below 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. Please refer to the cXML User Guide at www.cxml.org for more information.

Valid cXML interactions

Business Integration Connect supports the following cXML document flow definition interactions:

Source	Target	Source Package	Target Package	Source Protocol	Target Protocol	Pass Through	Validation	Translation
Particpant	Manager	None	None	cXML	cXML	x	x	
Manager	Participant	None	None	cXML	cXML	x	x	
Manager	Participant		None	XML	cXML	x	x	x

Creating a cXML document flow definition

Use the following process to create a new document flow definition for a cXML document.

Note: You must ensure that the correct version of cXML is defined before you create a cXML document flow definition. The default is version 1.2.009.

1. Click **Hub Admin** > **Hub Configuration** > **Document Flow Definition**.
2. Click **Create Document Flow Definition**. The Console displays the Create Document Flow Definitions screen.
3. Select **Document Flow** for Document flow type.
4. Enter either the request type, such as *OrderRequest*, in the **Code** and **Name** boxes. For Response document, if the Response does not have any child tags other than <Status>, enter *Response*; otherwise enter the next tag name following <Status>.
   For example:

```
<cXML>
    <Response>
        <Status code="200" text="OK"/>   --> The DocumentFlow code
    </Response>
</cXML>
<cXML>
    <Response>
        <Status code="200" text="OK"/>
   <ProfileResponse>   --> The DocumentFlow code
    </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 a **Description**.
7. Select **Yes** for **Document level**.
8. Select **Enabled** for **Status**.
9. Select **Yes** for all **Visibility** attributes.
10. Click on the **Package: None** folder to expand the package selection options.
11. Select the Protocol: cXML (1.2.009): cXML.
12. Click **Save**.

Once the document flow definition is created, enable the participant connections as needed. See “Activating participant connections” on page 58 for more information.

Notices and Trademarks

Notices

IBM may not offer the products, services, or features discussed in this document in all 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 give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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 program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM 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 measurement 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 necessarily 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.

This information may contain examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may 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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE

This information may contain sample application programs in source language, which illustrates 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.

Websphere Business Integration Connect contains code named ICU4J which is licensed to you by IBM under the terms of the International Program License Agreement, subject to its Excluded Components terms. However, IBM is required to provide the following language to you as a notice:

COPYRIGHT AND PERMISSION NOTICE

Copyright (c) 1995-2003 International Business Machines Corporation and others

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 may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: 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
the IBM logo
AIX
CrossWorlds
DB2
DB2 Universal Database
Domino
Lotus
Lotus Notes
MQIntegrator

MQSeries
Tivoli
WebSphere

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.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

WebSphere Business Integration Connect Enterprise and Advanced Editions includes software developed by the Eclipse Project (www.eclipse.org).

WebSphere Business Integration Connect Enterprise and Advanced Editions Version 4.2.2.

Printed in USA