WebSphere Adapters

WebSphere Adapter for Flat Files User Guide

Version 6.0

Note

Before using this information, be sure to read the general information in "Notices" on page 57.

7April2006

This edition of this document applies to version 6, release 0, of IBM WebSphere Adapter for Flat Files (product number 5724L78) and to all subsequent releases and modifications until otherwise indicated in new editions.

To send us your comments about this document, email 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

WebSphere Adapter for Flat Files Version 6.0 User Guide

The IBM® WebSphere Adapter for Flat Files facilitates the exchange of business objects between enterprise information system file systems and J2EE-based programming models.

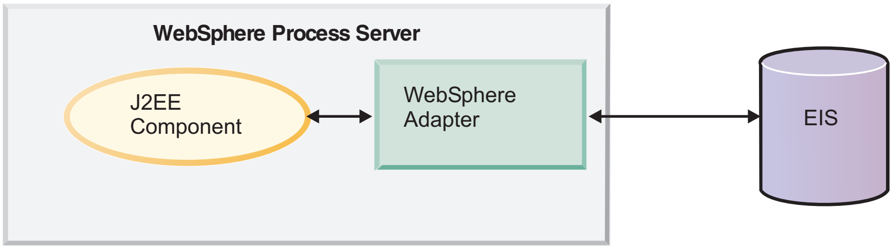
Product overview

This topic introduces the release, product features, and system requirements of the WebSphere Adapter for Flat Files.

IBM WebSphere Adapters

An IBM WebSphere Adapter implements the Java 2 Enterprise Edition (J2EE) Connector architecture (JCA), version 1.5. Also known as resource adapters or JCA adapters, WebSphere Adapters enable managed, bidirectional connectivity between enterprise information systems (EISs) and J2EE components supported by WebSphere Process Server.

A WebSphere Adapter



The IBM(R) WebSphere (R) Adapter portfolio is a new generation of adapters based on the Java 2 Platform, Enterprise Edition (J2EE) standard. JCA is a standard architecture for integrating J2EE applications with enterprise information systems. Each of these systems provides native APIs for identifying a function to call, specifying its input data, and processing its output data. The goal of the JCA is to provide an independent API for coding these functions, to facilitate data sharing, and to integrate J2EE applications with existing and other EISs. The JCA standard accomplishes this by defining a series of contracts that govern interactions between an EIS and J2EE components within an application server.

Fully compliant with the JCA standard, WebSphere Adapters have been developed to run on WebSphere Process Server. A WebSphere Adapter does the following:

- Integrates with WebSphere Process Server.
- Connects an application running on WebSphere Process Server with an EIS.
- Enables data exchange between the application and the EIS.

Each WebSphere Adapter is made up of the following:

- An implementation of the (J2EE) Connector Architecture (JCA), version 1.5 that supports WebSphere Process Server
- An enterprise metadata discovery component— you use this component with the enterprise service discovery wizard to introspect the EIS— to generate

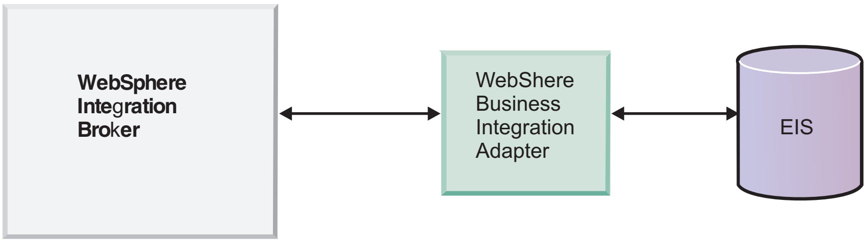
business objects and other service component architecture (SCA) artifacts that are compiled in a standard enterprise application archive (EAR) file.

WebSphere Adapters use service data objects (SDO) for representing data objects.

WebSphere Adapters and WebSphere Business Integration Adapters

Unlike WebSphere Adapters, WebSphere Business Integration Adapters are not JCA-compliant.

A WebSphere Business Integration Adapter


WebSphere Integration Broker
WebShere Business Integration Adapter
EIS

As shown in the figure, WebSphere Business Integration Adapters are distributed. They reside outside of the application server. The server, or integration broker, communicates with this type of adapter through a Java Messaging Service (JMS) transport layer.

Other differences between WebSphere Adapters and WebSphere Business Integration Adapters include the following:

- **Connection management** WebSphere Adapters rely on standard JCA contracts to manage life-cycle tasks such as stopping, starting; WebSphere Business Integration Adapters rely on the WebSphere Adapter Framework to manage connectivity.
- **Event notification** Known as inbound event notification for WebSphere Adapters.
- **Request processing** Known as outbound support in WebSphere Adapters.
- **Object definition** With WebSphere Adapters, you use an enterprise metadata discovery component to probe an EIS and develop business objects and other useful artifacts. This enterprise metadata discovery component is part of the WebSphere Adapter. WebSphere Business Integration Adapters use a separate Object Discovery Agent (ODA) to probe an EIS and generate business object definition schemas.

Audience

The information in this topic defines the users of the WebSphere® Adapter products and details the skills they require.

The audience for the adapter user guide includes data and application integrators who are responsible for assembling application components into a complete solution and preparing this solution for testing and deployment. These users require the following general skills:

- A good understanding of the business solution and the business environment
- Knowledge of application and solution components, to enable their efficient collaboration at run-time

- A detailed understanding of databases, data access issues, transactional models and connections across heterogeneous relational databases, queues, and web services
- Familiarity with integration tools

The application integrator is also responsible for detailed testing activities and needs these additional skills:

- Creating required scripts, tools, and templates for testing and deployment
- Creating integration workspaces and integrating systems & subsystems
- Resolving interdependencies between entities such as Enterprise Java Beans (EJBs), workflows, and web pages
- Validating the application or solution

The data integrator is also responsible for enabling access to a range of data sources for the application developers. The required skills include:

- Installing and configuring integration capabilities or point-to-point gateways
- Writing procedures to use database access logic efficiently
- Building data models for external data access tools
- Implementing security measures

Task road map: IBM WebSphere Adapter for Flat Files

The IBM WebSphere Adapter for Flat Files facilitates the exchange of business objects between flat files and J2EE-based programming models. The task road map gives the user the complete perspective from installing to the usage of WebSphere Adapter for Flat Files.

Task	Description
Learning about Business Objects	This topic describes the Business Objects for the WebSphere Adapter for Flat Files.
Installing the adapters	This topic describes how to install the WebSphere Adapter for Flat Files.
Deploying the adapter for inbound operation	This topic describes how to deploy the WebSphere Adapter for Flat Files for inbound operations.
Deploying the adapter for outbound operations	This topic describes how to deploy the WebSphere Adapter for Flat Files for outbound operations.
Configuring properties	This topic describes how to configure the properties for the WebSphere Adapter for Flat Files.
Troubleshooting the adapter	This topic describes how to troubleshoot the WebSphere Adapter for Flat Files.
Using the sample application	This topic describes how to use the sample applications for the WebSphere Adapter for Flat Files.

Differences between adapters

This topic describes the differences between the previous version of the WebSphere Business Integration Adapter for JText and the new WebSphere Adapter for Flat Files.

There are two main differences between the previous version of the WebSphere Business Integration Adapter for JText and the new WebSphere Business Integration Adapter for Flat Files. The architectural and functional differences are listed below.

- **Architectural differences**
  - **Protocol split** - With the WebSphere Business Integration Adapter for JText, the adapter combined the functionalities of both the local file operations and FTP file operations. With the WebSphere Adapter for Flat Files, the adapter only deals with local file operations. This functional segregation makes the adapter more maintainable and more specialized for protocol-specific operations.
  - **Perception of an inbound event** - With the WebSphere Business Integration Adapter for JText, individual data records in the event file were considered to be events and the adapter parsed the event files to extract the record content. With the WebSphere Adapter for Flat Files, the entire file is considered an event and the adapter does not parse the event file. This architecture was chosen in order to separate the different tasks involved with both file handling and data transformation. File handling deals with the entire file, which involves detecting the arrival of event files for inbound processing, reading the entire content of the file, and writing the file content for outbound processing. While data transformation involves file parsing and extracting data records out of the file. This split between protocol handling and data transformation makes the individual components more reusable and maintainable.
- **Functional differences**
  - **More outbound processing functionalities** - The WebSphere Adapter for Flat Files supports more operations for outbound processing compared to the WebSphere Business Integration Adapter for JText. The WebSphere Business Integration Adapter for JText request operations supported only create, append, and overwrite through the appropriate meta-object configuration. While the WebSphere Adapter for Flat Files supports create, append, overwrite, retrieve, delete, exists, and list operations.
  - **File splitting feature available for inbound processing** - With the WebSphere Adapter for Flat Files, an event file can be delivered to an endpoint in specified chunk sizes.

Enterprise service discovery

The enterprise service discovery wizard allows you to generate business objects for enterprise information system (EIS) or database entities.

The enterprise service discovery wizard provides a blue print for business objects. It allows you to browse the metadata information of an EIS or database, enables selection of the artifacts of interest, and generates deployable service objects and descriptions. By selecting meta-object nodes from the metadata tree structure, you can generate business objects for EIS or database entities. The metadata is transformed into service data objects consisting business graphs and business objects.

The enterprise service discovery wizard allows you to perform the following actions:

- Generate business objects
- Set application-specific information on the business objects
- Set application-specific information on properties

- Provide service descriptions for inbound and outbound events
- Provide connection descriptions for inbound and outbound events

How the adapter works

This topic describes how WebSphere Adapter for Flat Files works.

The IBM WebSphere Adapter for Flat Files is a JCA 1.5 specification compliant J2EE component, that enables bidirectional connectivity and access to files in an Enterprise Information System (EIS) file system. The adapter essentially allows J2EE clients access to any backend EIS applications that communicate only through files. The EIS file system can be the local file system mounted on the operating system on which the adapter runs or it may also be a mapped drive which is accessible from the local file system.

The events and responses are captured as files on the file system. The event files can be sent to an endpoint configured to receive the events. The requests can be initiated from any J2EE client and results in a response returned to the client after the request is processed.

The adapter has two modes of operation:

- Inbound event processing mode, and
- Outbound request processing mode

Inbound event processing

This topic describes the inbound event processing for WebSphere Adapter for Flat Files.

Inbound event processing

The inbound event processing is an asynchronous operation. The backend enterprise information system (EIS) generates events in the form of files. These files are stored in a user configured event directory. In the inbound processing mode, the adapter polls event files from a user configured event directory at regular intervals. When an event file arrives in the event directory, the adapter reads the entire event file as bytes, wraps the bytes inside a business object, and posts the business object to a subscribed endpoint. After the event file is posted, the adapter either archives the polled event file in an archive directory or deletes it based on the user configuration. The event directory, archive directory, the poll interval, and the poll quantity (the number of event files to poll in a single poll cycle) are all configurable parameters.

File Splitting

The inbound event processing mode also supports a file splitting feature, where the event file is split into several chunks and each chunk is posted to the endpoint separately. This reduces the memory loading during event processing. The client specifies the file split threshold and file chunk size in the adapter configuration. The properties FileSplitThreshold and FileChunkSize are configured in the ActivationSpecification. Any event file which exceeds the file split threshold is split into chunks whose size is as specified in the file chunk size. The event file splitting does not change the notion of poll quantity. The adapter still sends a maximum of poll quantity event files to the endpoint. However, now the event files which exceed the threshold size are sent to the endpoint as separate chunks. If FileSplitThreshold has a negative value, the chunking feature is disabled and the

FileChunkSize value is not considered. In this mode, the adapter can be configured either from WebSphere Process Server administrative console or using WebSphere Integration Developer.

When chunking is enabled, each chunk results in a business object. This means that the PollQuantity and the number of business objects obtained by the endpoint can be different.

For merging of files, the adapter does not take the ownership of reassembling the chunked data. Rather, it gives the information about the chunk so that an external application can merge the chunks. The chunk information is included in the outputString property of the business object. The chunk information includes the chunk size in bytes and the event ID.

Event management architecture and control flow

The event management framework (EMF) maintains information about the endpoint, which receives business objects from the adapter. The EMF internally uses the event data table (EDT) to track the events. The ActivationSpecification properties EDTDatabaseName, EDTDriverName, EDTTableName, EDTUserName and EDTUserPassword determine the configuration values for the event data table used by the event management framework. At the specified poll period, the adapter polls the files in the event directory, which are in accord with the user-configured file mask and poll quantity specification. The adapter internally uses an event table, addressed as Flat Files Event Table, to log the status of the events that have been polled but not yet posted to the endpoint. The ActivationSpecification property, FFEventTableName, determines the name of the Flat Files event table. FFEventTable is only created in the Cloudscape™ database and not in any database that is used for the EDT table. Once an event is polled, the adapter generates the Event ID and stores the event reference in the Flat Files event table with a 'NEW' status. The adapter then waits for the base class method callback to proceed further. The base class functions call back the adapter methods to process the events. The adapter changes the status for the event in the Flat Files event table to a 'IN_PROGRESS' status. It wraps the file content in a business object and posts the same across to the configured endpoint. The event entry is then deleted from the Flat Files event table. The event is archived optionally, based on the configuration.

Outbound request processing

This topic describes the outbound request processing for the WebSphere Adapter for Flat Files.

For outbound request processing, an external J2EE client can invoke the adapter through the Service Component Architecture (SCA). The client requests a connection, which in turn is passed from the adapter to the enterprise information system (EIS). Outbound support allows a client to make calls to the adapter to perform specific operations in an EIS file system.

Operations supported by outbound request processing mode include:

- Create
- Append
- Delete
- Retrieve
- Overwrite
- Exist

- List

Outbound request processing consists of the following steps:

1. The outbound service client looks up the business object factory service.
2. The outbound service client creates a business object from the business object factory.
3. The outbound service client locates the adapter service.
4. The outbound service client invokes the appropriate function on the adapter service by passing the function name and the business object.

Note: The adapter supports only synchronous outbound communication and does not support asynchronous mode.

Parameter passing in the SCA framework

Using the service client, the user can pass protocol-specific parameters, such as the directory path and file name, in the following ways:

1. Through the business object that is passed as a parameter while invoking the adapter service.
2. The parameters can also be set in WebSphere Integration Developer while generating the service artifacts. The parameters passed through the business object will override these values. Both directory path and file name should be set together through one or both of the above ways.

Locale and globalization support

This adapter has been globalized so that it can support single- and double-byte character sets and deliver message text in the specified language.

This adapter supports the processing of bidirectional script data for Arabic and Hebrew languages. To use the bidirectional capacity, you must configure the bidirectional properties. In this user guide, the term *bidirectional properties* refers to the properties that control invocation of bidirectional support.

If your enterprise information system (EIS) uses a bidirectional format that differs from the Windows® standard format, all properties with bidirectional support are transformed from the Windows standard format to the bidirectional format of the target EIS. The adapter also transforms such data from the EIS into Windows standard format before passing it to WebSphere Process Server.

The Java(TM) runtime environment within the Java Virtual Machine (JVM) represents data in the Unicode character code set. Unicode contains encodings for characters in most known character code sets (both single- and multi-byte). Most components in the WebSphere Business Integration system are written in Java. Therefore, when data is transferred between most WebSphere Business Integration system components, there is no need for character conversion.

To log error and informational messages in the appropriate language and for the appropriate country or territory, the adapter uses the locale of the system on which it is running.

WebSphere Process Server bidirectional language format

WebSphere Process Server uses the bidirectional language format of ILYNN (implicit, left-to-right, on, off, nominal), which is also the Windows bidirectional language format. All other bidirectional language formats must be converted prior to being introduced to WebSphere Process Server.

Five attributes must be set for the proper bidirectional language format. The attributes and settings are listed in the table titled “Bidirectional attributes.”

Bidirectional attributes

Letter position	Purpose	Values	Description	Default setting
1	Order Schema	I or V	Implicit (Logical) or Visual	I
2	Direction	L R C D	Left-to-Right Right-to-Left Contextual Left-to-Right Contextual Right-to-Left	L
3	Symmetric Swapping	Y or N	Symmetric Swapping is on or off	Y
4	Shaping	S N I M F B	Text is shaped Text is not shaped Initial shaping Middle shaping Final shaping Isolated shaping	N
5	Numeric Shaping	H, C, or N	Hindi, Contextual, or Nominal	N

The adapter is responsible for transforming data into a Logical-Left-to-Right format before sending the data into WebSphere Process Server components.

Note: The locale setting of the user interface (browser) defines the bidirectional language display and edit format. WebSphere Process Server user interfaces must convert locale-specific formats to the WebSphere Process Server default format.

Bidirectional property levels

You can set bidirectional properties at several different levels. For more details on these properties and how to set them using the enterprise service discovery wizard, refer to the sections on creating the adapter project and configuring the adapter.

Editing bidirectional properties

You can edit the bidirectional properties for business objects and business object attributes using annotations in the Business Object Editor in WebSphere Integration Developer. The annotations are stored in the business object (the *.xsd file). For more information, refer to the Business Object Editor documentation on the WebSphere Integration Developer website at http://www.ibm.com/software/integration/wid.

You can also edit certain bidirectional properties once they have been defined by using the assembly editor in WebSphere Integration Developer. For more information on using bidirectional properties at run time, refer to the general technical paper and the adapter technical paper regarding bidirectional support. For more information on the assembly editor, refer to the assembly editor documentation on the WebSphere Integration Developer website at http://www.ibm.com/software/integration/wid.

Business objects overview

This set of topics describes the business object naming conventions, structure, and supported operations for WebSphere Adapter for Flat Files.

Business object naming conventions

This topic describes the business object naming conventions for WebSphere Adapter for Flat Files.

For the Flat Files Adapter, there are two defined business objects: FlatFile.xsd and FlatFileBG.xsd. The business object structure is the same for both inbound and outbound operations. The business object schema, FlatFile.xsd, consists of the following attributes:

- directoryPath
- fileName
- inputBytes
- outputBytes
- outputString

FlatFileBG.xsd is the business graph that contains the attributes described above.

Business object structure

This topic describes the business object structure for WebSphere Adapter for Flat Files.

The Flat Files adapter business object structure is based on the generic WebSphere Business Integration business object structure, which is modeled as a base XML schema. The adapter has the same business object structure for both inbound and outbound events.

The ″Flat Files business object structure″ table defines the business object structure used during both inbound event processing and outbound request processing to transfer data to and from the enterprise information system.

Flat Files Business object structure

Attribute name	Attribute type	Description
directoryPath	String	Absolute path for the directory corresponding to the output file.
filename	String	Name of the event/output file with extension.
inputBytes	byte[]	Content of the file as-is, which would be passed from/to the Flat Files resource adapter.
outputBytes	byte[]	Contains the output bytes for operations. For example, retrieve.

Attribute name	Attribute type	Description
outputString	String	Contains the output of various outbound operations. For example, list.

Attribute properties

This topic describes the attribute properties for WebSphere Adapter for Flat Files.

The ″Attribute properties″ table below defines the attribute properties for the Flat Files adapter.

Attribute properties

Attribute property	Description
Cardinality	Only single cardinality flat business objects are supported.
Key and foreign key	The key for the business object structure, both on request and response, is a combination of DirectoryPath and Filename attributes. It is the same for all operations. Foreign keys are not supported.
Name	Indicates the name of the business object attribute.
Required	Indicates that the attribute field is a required field. Since the key is a combination key, both the DirectoryPath and Filename attributes cannot be null. One null value is valid. However, one non-null value is required.
Special	None.
Type	Indicates the type of the business object attribute. The type can be a string or complex type representing an integration object.

Supported operations

The WebSphere Adapter for Flat Files performs inbound and outbound operations. The supported operations and the business object structures for the supported operations are listed here.

The ″Supported operations″ table below describes the operations supported by the Flat Files adapter.

Note: The WebSphere Adapter for Flat Files does not support verbs. The adapter only supports operations.

Supported operations

Operation	Response
Append	The content in the request is appended at the end of the file.

Operation	Response
Create	File with the specified filename is created in the specified directory with the content sent across in the request.
Delete	Deletes the file from the directory specified in the request.
Exists	If the file in the request exists in the specified directory, true is returned in outputString field, else false is returned in the outputString.
List	The operation would return all the filenames in the directory specified in the request.
Overwrite	Overwrites the file in the directory with the content specified in the request.
Retrieve	Response returns the file content of the filename specified in the request.

The ″Business object structure for Create″ table below defines the business structure for the Create operation.

Business object structure for Create

<table>
<tr><th>Name request</th><th>Value request</th><th>Name response</th><th>Value response</th></tr>
<tr><td>DirectoryPath</td><td>/home/user/ outputdir</td><td>DirectoryPath</td><td>/home/user/ outputdir</td></tr>
<tr><td>Filename</td><td>Myoutput.out</td><td>Filename</td><td>Myoutput.out</td></tr>
<tr><td>InputBytes</td><td>MSH|^~\& ||.||| 199908180016|| ADT^A04| ADT.1.1698593|P</td><td>InputBytes</td><td>MSH|^~\& ||.||| 199908180016|| ADT^A04| ADT.1.1698593|P</td></tr>
<tr><td>OutputBytes</td><td>Null</td><td>OutputBytes</td><td>Null</td></tr>
<tr><td>OutputString</td><td>Null</td><td>OutputString</td><td>Null</td></tr>
</table>

The ″Business object structure for list″ table below defines the business structure for the List operation.

Business object structure for List

Name request	Value request	Name response	Value response
DirectoryPath	/home/user/ outputdir	DirectoryPath	/home/user/ outputdir
Filename	Null	Filename	Null
InputBytes	Null	InputBytes	Null
OutputBytes	Null	OutputBytes	Null
OutputString	Null	OutputString	File01.out, File02.out, File03.out, File04.out

The ″Business object structure for Exists″ table below defines the business structure for the Exists operation.

Business object structure for Exists

Name request	Value request	Name response	Value response
DirectoryPath	/home/user/outputdir	DirectoryPath	/home/user/outputdir
Filename	File01.out	Filename	File01.out
InputBytes	Null	InputBytes	Null
OutputBytes	Null	OutputBytes	Null
OutputString	Null	OutputString	True

Installing the adapter

This topic introduces the task of installing the WebSphere Adapter for Flat Files.

For information about installing the adapter, see Installing IBM WebSphere Adapters.

Adapter environment

This topic introduces information about the environment for the WebSphere Adapter for Flat Files.

Hardware and software requirements

For hardware and software requirements for this adapter, see IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: Hardware and Software Requirements. Select your adapter from the list of WebSphere adapters.

Adapter-specific installation information

Before you install WebSphere Adapter for Flat Files, there are a few additional installation requirements.

Inbound operations

Before you install WebSphere Adapter for Flat Files, make sure the following is completed for inbound operations:

1. The Event Directory specified must exist in the file system and match the one specified in the ActivationSpecification configuration).
2. If archiving is enabled, the ArchiveDirectory specified must exist in the file system.
3. The Event Data Table database parameters specified should be proper, and corresponding database entities must exist.
4. If the Event Data Table database is not the Cloudscape database included with WebSphere Process Server, the database driver must be in the CLASSPATH of the adapter.
5. If the FFEventTable is created by the user (and not the adapter), the FFEventTable should have the correct schema. Also the FFEventTable should be used exclusively by the adapter and not any other application.

   Note: The Flat Files resource adapter-specific database is created by the adapter only on Cloudscape.

Outbound operations

If the OutputDirectory is specified, make sure it exists in the file system.

Installed file structure

The WebSphere Adapter for Flat Files installation provides a Resource Adapter Archive (RAR) file that is deployable on WebSphere Process Server. The RAR files contains the files that are shipped with the adapter.

The RAR file for the WebSphere Adapter for Flat Files is delivered during the installation process. Files that are shipped with the adapter package are preceded with ′CWYFF′, which is the component ID allocated for the WebSphere Adapter for Flat Files. The table ″Packaging structure″ describes the contents of the files contained in the RAR file.

Packaging structure

File	Description
CWYBS_AdapterFoundation.jar	This is the foundation class jar file.
CWYFF_FlatFile.jar	This contains the Flat Files Resource Adapter-specific subclasses.
icu4j_3_2.jar	The ICU4J libraries required for globalization.
meta-inf\Manifest.mf	
meta-inf\discovery-service.xml	This is used by the enterprise service discovery wizard.
meta-inf/ra.xml	This is the deployment descriptor.
Flatfile.xsd	This is packaged under Dependencies. It contains a sample of the WebSphere business object structure required by the Flat Files Resource Adapter.

Deploying the adapter for inbound operations

After you install the WebSphere Adapter for Flat Files, you must deploy it. Deployment consists of creating a project, generating service artifacts, generating reference bindings, exporting the application, installing the application to run on WebSphere Process Server.

The adapter is distributed as a Resource Adapter Archive (RAR) file.

1. You install the adapter into WebSphere Integration Developer by importing the RAR file.
2. Once the adapter has been installed, you generate an Enterprise Application Archive (EAR) file.
3. The EAR file is then deployed to WebSphere Process Server using the administrative console.

Note: While WebSphere Integration Developer only runs on Windows or Linux®, WebSphere Process Server runs on Windows, Linux and UNIX® platforms.

In principle, deploying the adapter is the same as deploying any other component on WebSphere Process Server. For more information on deploying components on WebSphere Process Server, see the IBM WebSphere Integration Developer, Version 6.0 documentation.

Deployment prerequisites

You must install these products before you can deploy the adapter:
- WebSphere Integration Developer V6.0 (WebSphere Integration Developer)
- WebSphere Adapter for Flat Files, installed on the same machine as WebSphere Integration Developer
- WebSphere Process Server administrative console

For WebSphere Process Server installation instructions, see the IBM WebSphere Process Server for Multiplatforms, Version 6.0 documentation.

Creating a project for the adapter for inbound operations

The first step of deploying the adapter is to import the adapter RAR file that was installed during installation into the WebSphere Integration Developer. Importing the RAR file automatically creates a new project.

The following steps are performed using WebSphere Integration Developer. For details about this tool, refer to the IBM WebSphere Integration Developer, Version 6.0 documentation.

1. Launch the WebSphere Integration Developer.
2. Switch to the J2EE perspective.
3. To import the RAR file into WebSphere Integration Developer, right-click the window frame of the J2EE perspective and select **File** → **Import** from the pop-up menu.
4. Select RAR file and click **Next**. The Connector Import window opens.
5. Select the location from where you will import the RAR file (the same location where you copied your adapter file during installation), and specify a project name.
6. Deselect the **Add module to an EAR project** check box.
7. Click **Finish** to import the RAR file. This creates a new J2EE Connector project in the workspace.

Now that you have created a new adapter project by importing the RAR file into WebSphere Integration Developer, you are ready to use enterprise service discovery wizard to generate service artifacts.

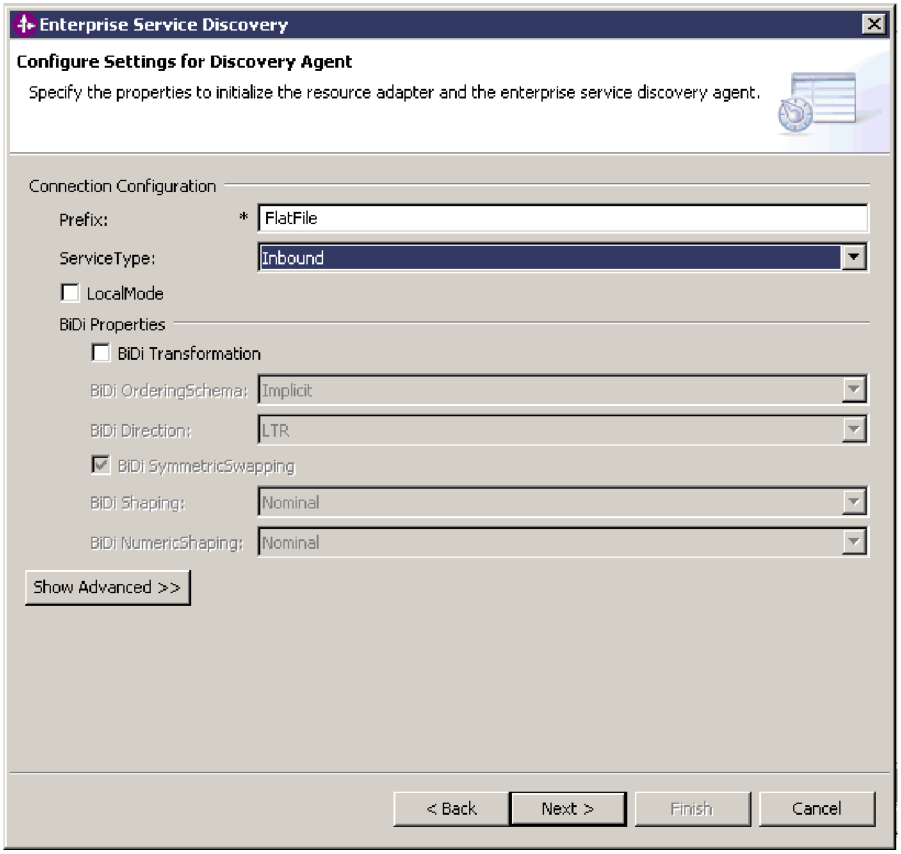
Generating service artifacts for inbound operations

The generation of service artifacts for inbound operations is done using the enterprise service discovery wizard in WebSphere Integration Developer.

As you complete the process of generating service artifacts for inbound operations, you will enter all the information necessary to configure the adapter for the first time. The output from the enterprise service discovery wizard is saved to a business integration module, which contains the business objects, the import file (which describes configuration for outbound processing, as defined by the

ConnectionFactory), the export file (which describes configuration for inbound event processing, as defined by the ActivationSpecification), and the Web Services Description Language (WSDL) file.

1. From the WebSphere Integration Developer window, switch to the Business Integration perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select Business Integration perspective in WebSphere Integration Developer.
2. In the Business Integration Perspective of WebSphere Integration Developer, right-click the frame and from the pop-up menu, select **New** → **Enterprise Service Discovery**. The Select an Enterprise Service Resource Adapter window is displayed.
3. Select the **IBM WebSphere Adapter for Flat Files (version 6.0.0) from the 'CWYFF_FlatFile' Connector Project** and click **Next**.
4. In the Configure Settings for Discovery Agent window, as illustrated in the graphic below, set the connection properties by selecting Inbound for the ServiceType and click **Next**.



Enterprise Service Discovery

Configure Settings for Discovery Agent

Specify the properties to initialize the resource adapter and the enterprise service discovery agent.

Connection Configuration

Prefix: * FlatFile

ServiceType: Inbound

☐ LocalMode

BiDi Properties

☐ BiDi Transformation

BiDi OrderingSchema: Implicit

BiDi Direction: LTR

☑ BiDi SymmetricSwapping

BiDi Shaping: Nominal

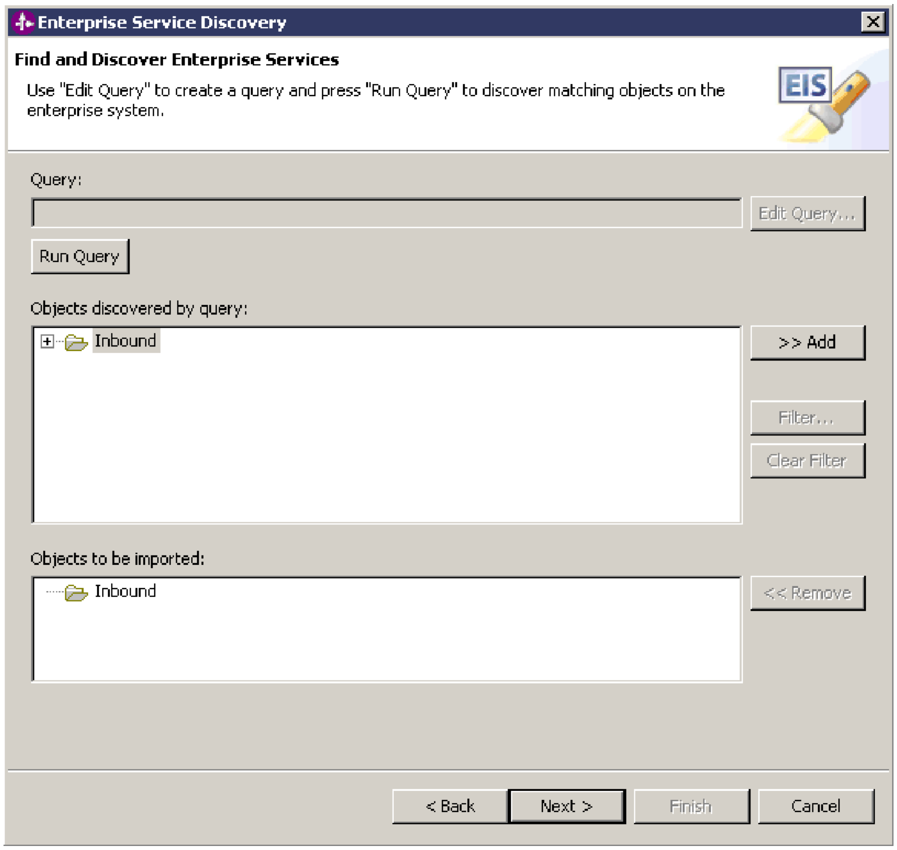
BiDi NumericShaping: Nominal

The Configure Settings for Discovery Agent window

5. (Optional) To enable BiDi support, select the **BiDi Transformation** check box in the Configure Settings for Discovery Agent window and select the appropriate BiDi format.
6. (Optional) At the bottom of the Configure Settings for Discovery Agent window, click the Show Advanced button. You can set logging and tracing levels here. Click **Next**.
7. In the Find and Discover Enterprise Services window, click the **Run Query** button to display the meta-data tree of the Flat Files adapter.
8. Under Objects discovered by query, for inbound operations, select the Inbound root node of the meta-data tree. As illustrated in the graphic below,

the objects that can be selected for import are displayed in the Objects discovered by query box. Highlight the object you would like to import, then click the **Add Selected** button to add the objects to the Objects to be imported box. Click **Next**.

Note: To remove objects from the Objects Discovered by query text box, highlight the object that you would like to remove and click the **Remove Selected** button.



Enterprise Service Discovery

Find and Discover Enterprise Services

Use "Edit Query" to create a query and press "Run Query" to discover matching objects on the enterprise system.

Query:

Objects discovered by query:

Inbound

Objects to be imported:

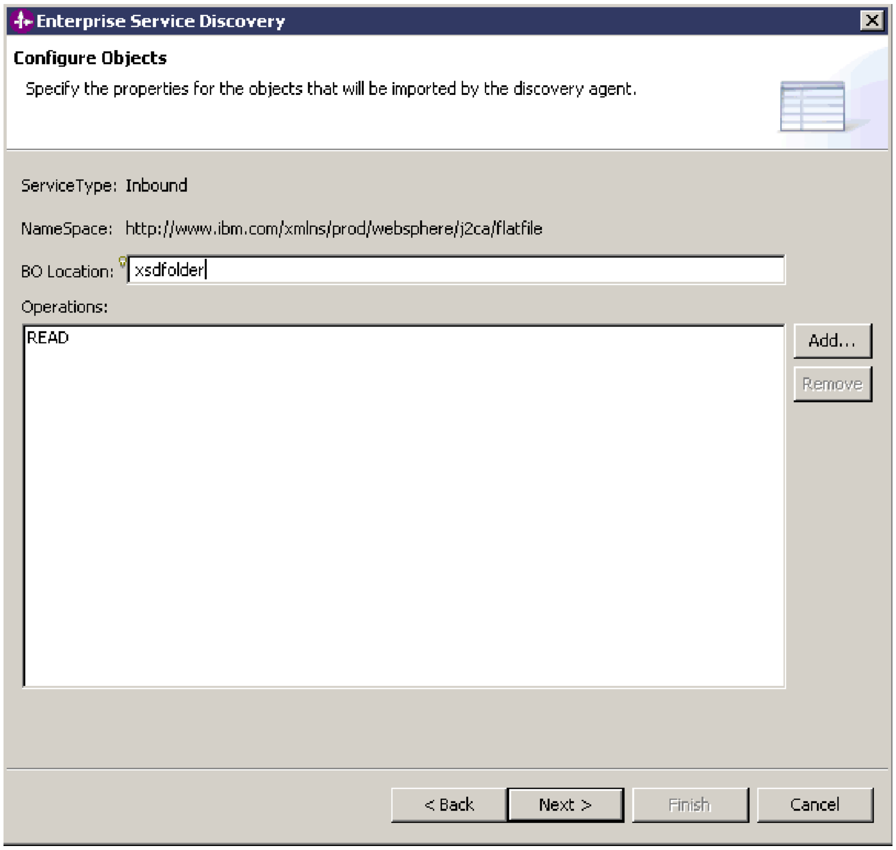
Inbound

The Find and Discover Enterprise Services window

9. In the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent. As illustrated in the graphic below, for the BO Location property, specify the folder name within the Business Integration module where the .xsd file should be generated. Click **Next**.

   The supported operations for Inbound include:

   - Read



Configure Objects

Specify the properties for the objects that will be imported by the discovery agent.

ServiceType: Inbound

NameSpace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile

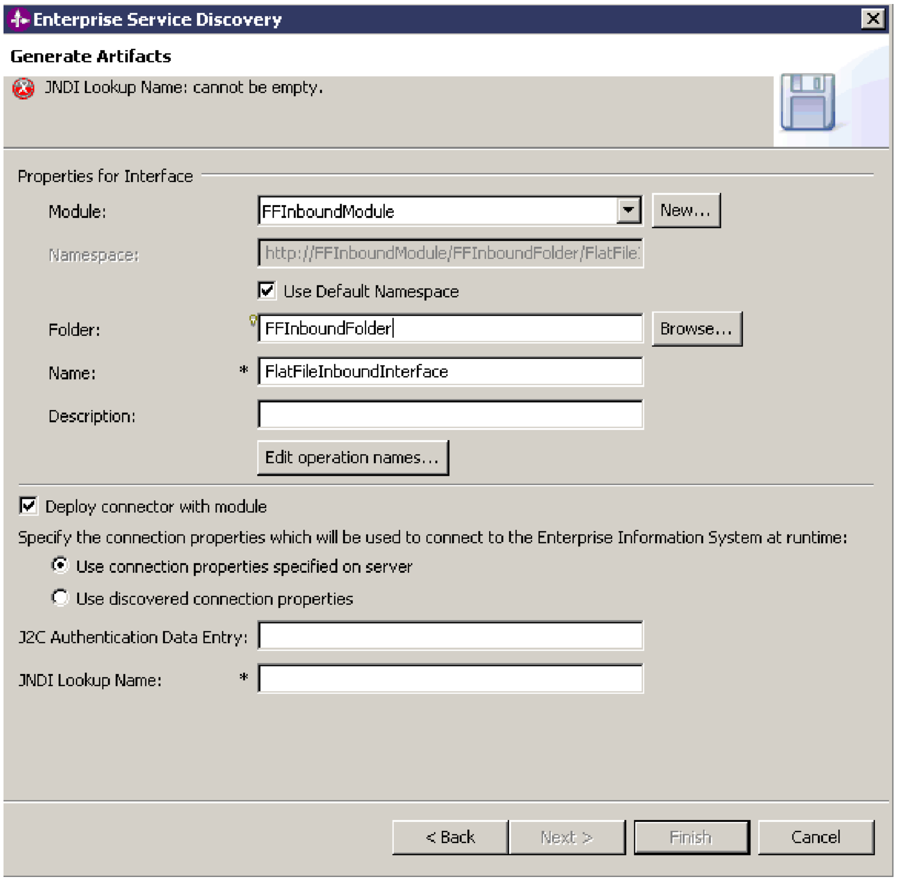
BO Location: xsdfolder

Operations:

READ

The Configure Objects window

10. In the Generate Artifacts window, do the following:
    a. Click **New** to create a new business integration module.
    b. In the New Module window, enter the Module name and click **Finish**.
    c. In the Generate Artifacts window, enter the folder name.
    d. As illustrated in the graphic below, under ″Specify the connection properties which will be used to connect to the Enterprise Information System at runtime″, click**Use discovered connection properties**.



Enterprise Service Discovery

Generate Artifacts

JNDI Lookup Name: cannot be empty.

Properties for Interface

Module: FFInboundModule

Namespace: http://FFInboundModule/FFInboundFolder/FlatFile

☑ Use Default Namespace

Folder: FFInboundFolder

Name: * FlatFileInboundInterface

Description:

☑ Deploy connector with module

Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

● Use connection properties specified on server

○ Use discovered connection properties

J2C Authentication Data Entry:

JNDI Lookup Name: *

The Generate Artifacts window

e. Enter the Event Distribution Database properties.

   Note: Properties marked with an asterisk (*) are required.

f. Enter the Flat Files adapter ActivationSpecification properties.

   Note: Properties marked with an asterisk (*) are required.

g. Enter the logging and tracing values.

   Note: Properties marked with an asterisk (*) are required.

h. (Optional) Enter the BiDi properties.

   Note: Clear the**Turn BiDi Off** check box, otherwise BiDi support will not be invoked.

11. When you are finished entering property values, click **Finish**.

Now, you can use WebSphere Integration Developer to generate reference bindings for the SCA component that the enterprise service discovery wizard creates.

Related tasks

"Enabling logging" on page 42
The WebSphere Adapter for Flat Files maintains a log file that you can view to determine the status of event processing. All events and errors that relate to the adapter are tracked by the log file, along with the date, time, and event for each log entry. Since the adapter logs an error message when it encounters an error or warning condition, the log file is a good source to start troubleshooting problems.

"Enabling tracing" on page 44
Tracing determines what level of errors or warnings are captured in the adapter log file. You can trace messages regarding adapter processing by defining a tracing level.

"Configuring properties" on page 34
You can configure adapter properties using the using WebSphere Process Server administrative console.

Related reference

"Custom properties" on page 38
This topic describes the default configuration properties for WebSphere Adapter for Flat Files.

"J2C activation specification properties" on page 36
This topic describes J2C activation specification properties (also referred to as message end point properties), which correspond to the ActivationSpecification interface of the J2EE Connector Architecture Specification.

"Resource adapter properties" on page 40
Using the enterprise service discovery wizard when you first configure the adapter (and later, via the WebSphere Process Server administrative console), you can configure Resource adapter properties. This category of properties includes logging and tracing properties and adapter-specific properties.

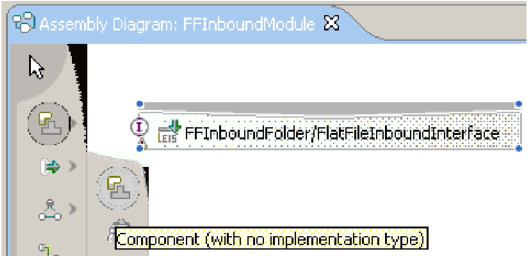
Generating reference bindings for inbound operations

After you have generated service artifacts, you can generate reference bindings using WebSphere Integration Developer.

Prerequisite: An adapter project must be created and configured on your workspace. Once an adapter project is created, you must generate reference findings to bind to the service component.

Reference bindings are used by other WebSphere Business Integration SCA components to access the adapter. You create a reference to the adapter from the project module so as to link the adapter to the other server processes.

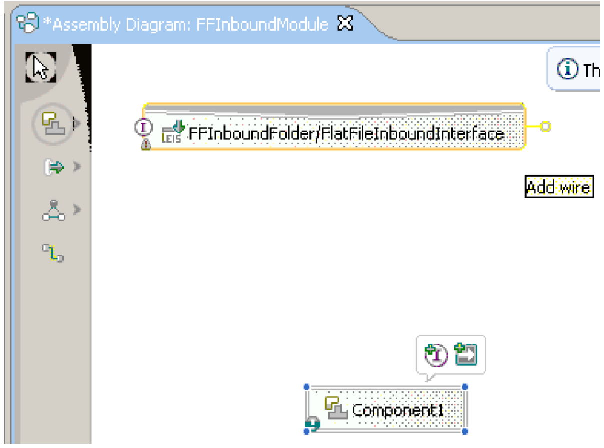
1. From WebSphere Integration Developer window, switch to the Business Integration perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select Business Integration perspective in WebSphere Integration Developer.
2. In the Business Integration Perspective of WebSphere Integration Developer, right-click the module, and select **Open With** → **Assembly Editor**. The Assembly Diagram window appears with the modules Import component in view.
3. To create a new component, click the top-most icon in the left-side (vertical) frame of the Assembly Diagram Window.


Assembly Diagram: FFInboundModule
FFInboundFolder/FlatFileInboundInterface
Component (with no implementation type)

The Assembly Diagram window

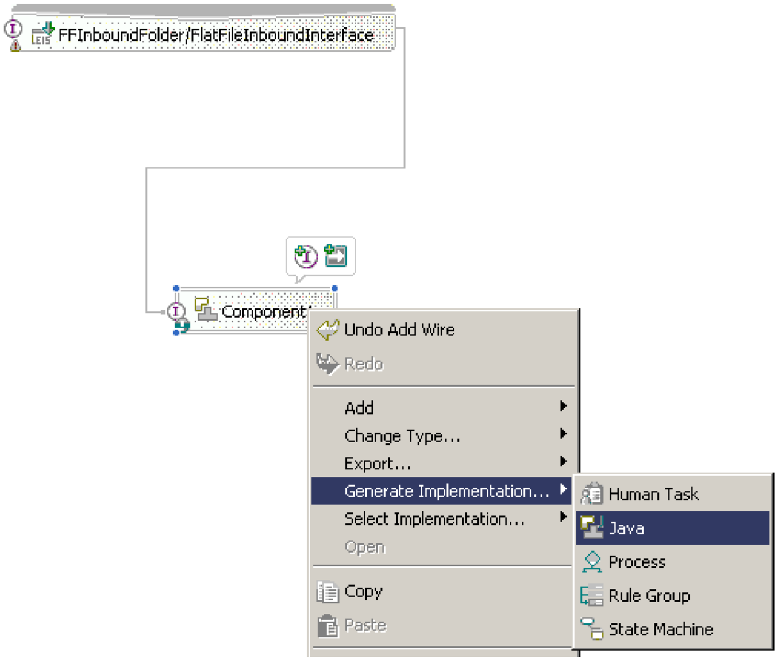
A new menu of icons appears. The graphic above illustrates the new menu of icons.

4. Create a Component with no implementation. As illustrated in the graphic below, drag the Create a Component icon into the Assembly Diagram window.


*Assembly Diagram: FFInboundModule
Th
FFInboundFolder/FlatFileInboundInterface
Add wire
Component1

The Assembly Diagram window

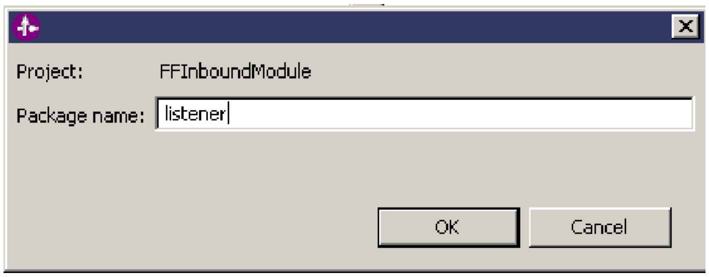
5. Wire the Component to the import module. Click and drag the module‘s Import component to the new component. This draws a wire from the Import component to the new component.
6. In the Add Wire dialog box, click **OK**. The new component displays in the Assembly Diagram window with a wire that connects it to the modules Import component.
7. Create a Java™ component that will acts like an endpoint listener. During event delivery of the inbound operation, the adapter invokes the READ method of the component implementation and passes the inbound business object as a parameter.
   a. As illustrated in the graphic below, right click the component and select **Generate implementation** → **Java**.



The Assembly Diagram window

b. Select the package where the Java code should be created and click **OK**.

c. As illustrated in the graphic below, enter the Package name when prompted and click **OK**.



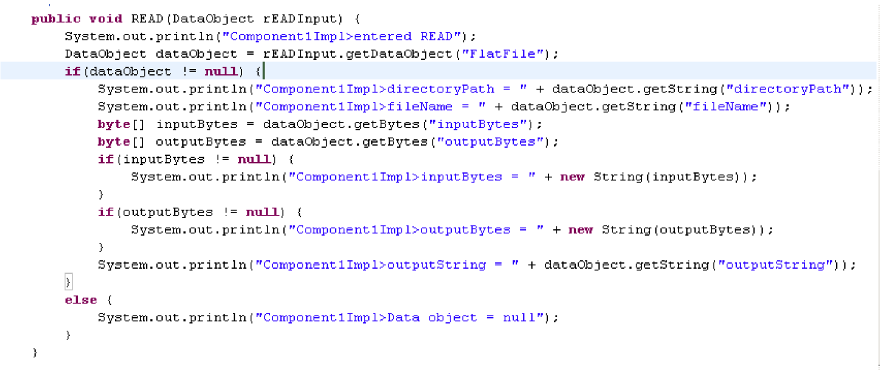
Project: FFInboundModule

Package name: listener

Package name dialog box

d. Click **OK** in the Generate Implementation window.

e. In the generated Java implementation, scroll to the READ method. As illustrated in the graphic below, you can add custom code inside the READ method to process the delivered business object according to your specific business needs.



```java
public void READ(DataObject rEADInput) {
    System.out.println("Component1Impl>entered READ");
    DataObject dataObject = rEADInput.getDataObject("FlatFile");
    if(dataObject != null) {
        System.out.println("Component1Impl>directoryPath = " + dataObject.getString("directoryPath"));
        System.out.println("Component1Impl>fileName = " + dataObject.getString("fileName"));
        byte[] inputBytes = dataObject.getBytes("inputBytes");
        byte[] outputBytes = dataObject.getBytes("outputBytes");
        if(inputBytes != null) {
            System.out.println("Component1Impl>inputBytes = " + new String(inputBytes));
        }
        if(outputBytes != null) {
            System.out.println("Component1Impl>outputBytes = " + new String(outputBytes));
        }
        System.out.println("Component1Impl>outputString = " + dataObject.getString("outputString"));
    }
    else {
        System.out.println("Component1Impl>Data object = null");
    }
}
```

Sample READ code

8. Save the file, **File** → **Save**.

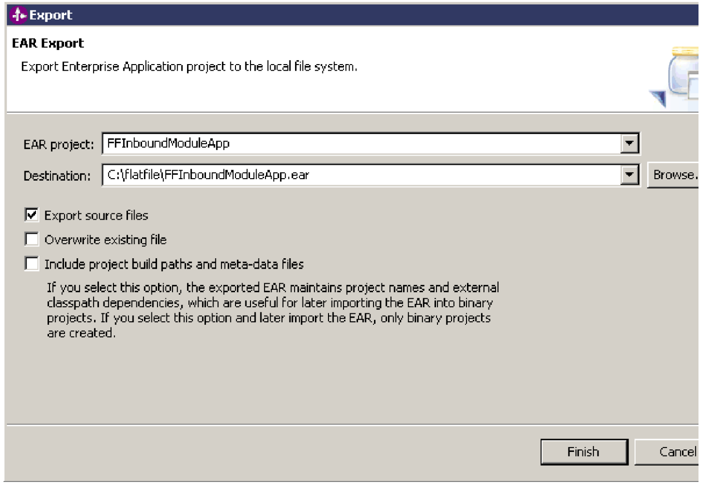
Now, you are ready to export the EAR file.

Exporting the EAR file for inbound operations

Before you can run the project, you must export it to an EAR file using WebSphere Integration Developer.

1. From WebSphere Integration Developer window, switch to the J2EE perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select J2EE perspective in WebSphere Integration Developer.
2. In the J2EE perspective window of the WebSphere Integration Developer tool, right click the module and select Export from the pop-up menu. The Export window is displayed.
3. Select EAR file from the Export - Select window. The Export - EAR Export window is displayed.
4. As illustrated in the graphic below, in the EAR Export window, select the EAR project and the destination directory (the directory, including the EAR file

name, where the project should be exported into).



Export
EAR Export
Export Enterprise Application project to the local file system.
Browse.
Finish
Cancel

EAR project: FFInboundModuleApp

Destination: C:\flatfile\FFInboundModuleApp.ear

☑ Export source files

☐ Overwrite existing file

☐ Include project build paths and meta-data files

If you select this option, the exported EAR maintains project names and external classpath dependencies, which are useful for later importing the EAR into binary projects. If you select this option and later import the EAR, only binary projects are created.

The EAR Export window

5. Click **Finish**.

Now that you have exported the project to an Enterprise Application Archive (.EAR) file, you are ready to install the application.

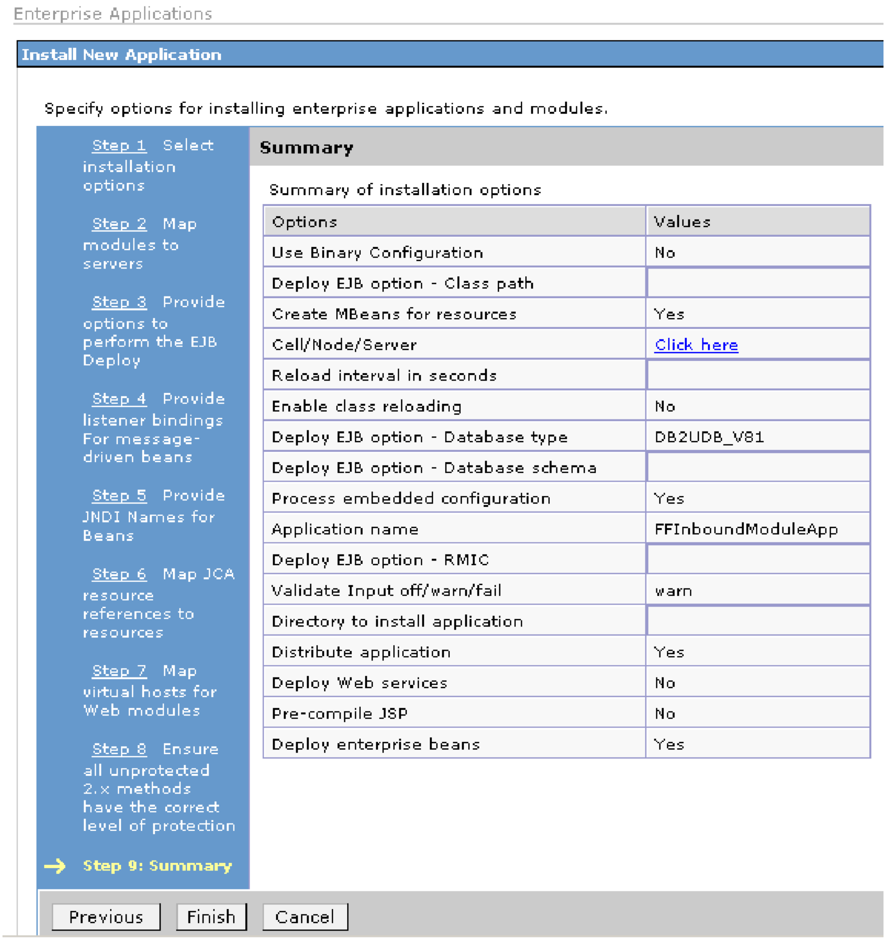
Installing the application for inbound operations

Installing the application project module is the last step of the deployment process.

When you install the application and run it, the adapter, which is embedded in the project module, runs as part of the installed application.

1. In the WebSphere Process Server administrative console, click **Applications** → **Install New Applications**.
2. Select the EAR file from the list of Enterprise Applications.
3. Under Path to the new application, specify the path of the EAR file, then click **Next**.
4. Continue to click Next through the various Step windows, until you reach the window titled with the step Map resource reference to resources.
5. A summary of all the installation options is displayed. As illustrated in the graphic below, in the Summary window, verify that all options are as you

intended and click **Finish**.


Enterprise Applications
Install New Application
Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide options to perform the EJB Deploy
Step 4 Provide listener bindings For message-driven beans
Step 5 Provide JNDI Names for Beans
Step 6 Map JCA resource references to resources
Step 7 Map virtual hosts for Web modules
Step 8 Ensure all unprotected 2.x methods have the correct level of protection
→ Step 9: Summary
Previous
Finish
Cancel

Specify options for installing enterprise applications and modules.

Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Deploy EJB option - Class path	
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Deploy EJB option - Database type	DB2UDB_V81
Deploy EJB option - Database schema	
Process embedded configuration	Yes
Application name	FFInboundModuleApp
Deploy EJB option - RMIC	
Validate Input off/warn/fail	warn
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	Yes

The WebSphere Process Server administrative console

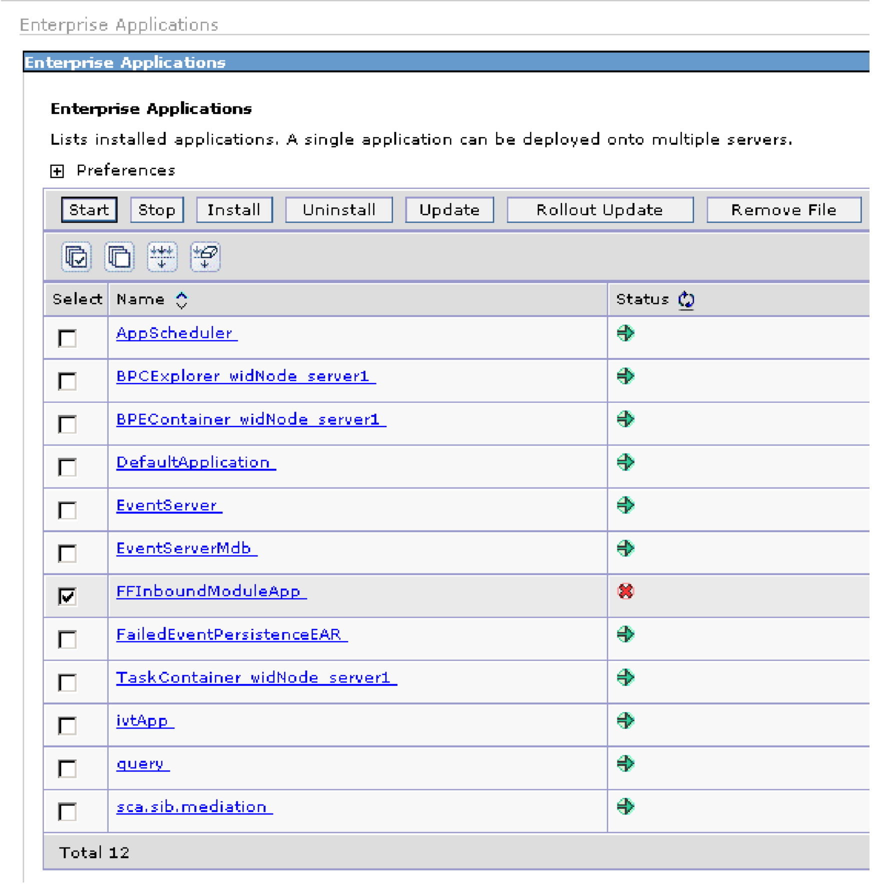
6. Click **Next**. Verify that all options are as you intended and click **Finish**.
7. A list of installation messages appears. Confirm that the message Application installed successfully is included at the end of the list.
8. Click the **Save to Master Configuration** link that appears at he end of the list of installation messages. The Enterprise Applications window is displayed.
9. Click **Save** to save the application. The application is now deployed and the Enterprise Applications window for the deployed application is displayed.

The application is now deployed and properly configured. The next step is to run the application.

Running the application for inbound operations

After you have deployed the application, you can run it. Since the adapter is embedded in the application project, when you run the application, the adapter is triggered to start running.

1. In the WebSphere Process Server administrative console, select **Applications** → **Enterprise Applications**.
2. As illustrated in the graphic below, select the check box for the application and click the **Start** button. The application starts running.



Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

Select	Name	Status
□	AppScheduler	➡
□	BPCExplorer_widNode_server1	➡
□	BPEContainer_widNode_server1	➡
□	DefaultApplication	➡
□	EventServer	➡
□	EventServerMdb	➡
☑	FFInboundModuleApp	✖
□	FailedEventPersistenceEAR	➡
□	TaskContainer_widNode_server1	➡
□	ivtApp	➡
□	query	➡
□	sca.sib.mediation	➡
Total 12		

The WebSphere Process Server administrative console

Deploying the adapter for outbound operations

After you install the WebSphere Adapter for Flat Files, you must deploy it. Deployment consists of creating a project, adding external dependencies to that project, configuring the service, and then deploying the application to run on WebSphere Process Server.

The adapter is distributed as a Resource Adapter Archive (RAR) file.

1. You install the adapter into WebSphere Integration Developer by importing the RAR file.
2. Once the adapter has been installed, you generate an Enterprise Application Archive (EAR) file.
3. The EAR file is then deployed to WebSphere Process Server using the administrative console.

Note: While WebSphere Integration Developer only runs on Windows or Linux, WebSphere Process Server runs on Windows, Linux and UNIX platforms.

In principle, deploying the adapter is the same as deploying any other component on WebSphere Process Server. For more information on deploying components on WebSphere Process Server, see the IBM WebSphere Integration Developer, Version 6.0 documentation.

Deployment prerequisites

You must install these products before you can deploy the adapter:

- WebSphere Integration Developer V6.0 (WebSphere Integration Developer)
- WebSphere Adapter for Flat Files, installed on the same machine as WebSphere Integration Developer
- WebSphere Process Server administrative console

For WebSphere Process Server installation instructions, see the IBM WebSphere Process Server for Multiplatforms, Version 6.0 documentation.

Creating a project for the adapter for outbound operations

The first step of deploying the adapter is to import the adapter RAR file that was installed during installation into WebSphere Integration Developer. Importing the RAR file automatically creates a new project.

The following steps are performed using WebSphere Integration Developer. For details about this tool, refer to the IBM WebSphere Integration Developer, Version 6.0 documentation.

1. Launch WebSphere Integration Developer.
2. Switch to the J2EE perspective.
3. To import the RAR file into WebSphere Integration Developer, right-click the window frame of the J2EE perspective and select **File** → **Import** from the pop-up menu.
4. Select RAR file and click **Next**. The Connector Import window opens.
5. Select the location from where you will import the RAR file (the same location where you copied your adapter file during installation), and specify a project name.
6. Deselect the **Add module to an EAR project** check box.
7. Click **Finish** to import the RAR file. This creates a new J2EE Connector project in the workspace.

Now that you have created a new adapter project by importing the RAR file into WebSphere Integration Developer, you are ready to use the enterprise service discovery wizard to generate service artifacts.

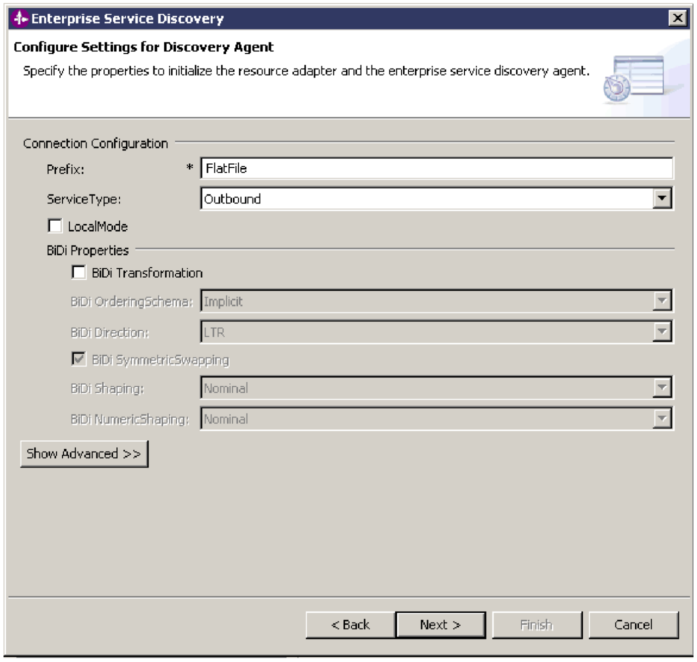
Configuring a service for outbound operations

The configuration process is done using the enterprise service discovery wizard in WebSphere Integration Developer.

As you complete the process you will enter all the information necessary to configure the adapter for the first time. The output from the enterprise service discovery wizard is saved to a business integration module, which contains the business objects, the import file (which describes outbound processing, as defined by the ActivationSpecification), the export file (which describes inbound event processing, as defined by the InteractionSpecification), and the Web Services Description Language (WSDL) file.

Important: During deployment, if you specify J2C activation specification properties when you initially configure the service, those property settings will remain in place (in other words, you cannot update the properties later, after you install the application, via the WebSphere Process Server administrative console - the properties can still be updated, but the adapter will not recognize the updated values if those properties are already set during deployment). If, for any reason, you want to set the J2C activation specification properties after installing the application via the administrative console, then you should refrain from setting them during deployment. Note that J2C connection factory properties can be set during deployment and then updated via the administrative console after you install the application.

1. From the WebSphere Integration Developer window, switch to the Business Integration perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select Business Integration perspective in WebSphere Integration Developer.
2. In the Business Integration Perspective of WebSphere Integration Developer, right-click the frame and from the pop-up menu, select **New** → **Enterprise Service Discovery**. The Select an Enterprise Service Resource Adapter window is displayed.
3. Select the **IBM WebSphere Adapter for Flat Files (version 6.0.0) from the 'CWYFF_FlatFile' Connector Project** and click **Next**. The Configure Settings for Discovery Agent window is displayed.
4. In the Configure Settings for Discovery Agent window, as illustrated in the graphic below, set the connection properties by selecting Outbound for the ServiceType. Outbound is the default ServiceType.



Enterprise Service Discovery

Configure Settings for Discovery Agent

Specify the properties to initialize the resource adapter and the enterprise service discovery agent.

Connection Configuration

Prefix: * FlatFile

ServiceType: Outbound

□ LocalMode

BiDi Properties

□ BiDi Transformation

BiDi OrderingSchema: Implicit

BiDi Direction: LTR

☑ BiDi SymmetricSwapping

BiDi Shaping: Nominal

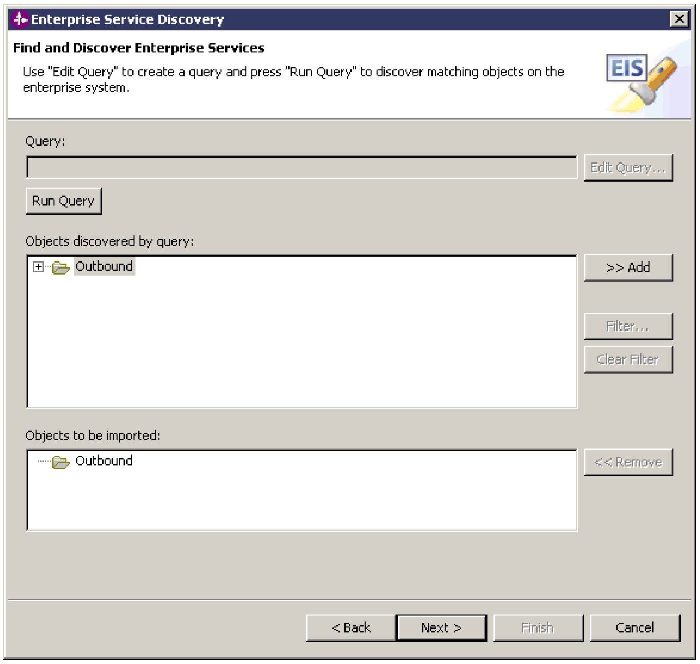
BiDi NumericShaping: Nominal

The Configure Settings for Discovery Agent window

5. (Optional) To enable BiDi support, select the **BiDi Transformation** check box in the Configure Settings for Discovery Agent window and select the appropriate BiDi format.
6. At the bottom of the Discovery Agent Initialize Properties window, click the Show Advanced button. You can set logging and tracing levels here. Click **Next**.
7. In the Find and Discover Enterprise Services window, click the **Run Query** button to display the meta-data tree of the Flat Files adapter.
8. Under Objects Discovered, select the Inbound root node of the meta-data tree. As illustrated in the graphic below, the objects that can be selected for import are displayed in the Objects Discovered box. Highlight the object you would

like to import, then click the **Add Selected** button to add the objects to the Objects to be imported box. Click **Next**.

Note: To remove objects from the Objects Discovered box, highlight the object that you would like to remove and click the **Remove Selected** button.



Enterprise Service Discovery

Find and Discover Enterprise Services

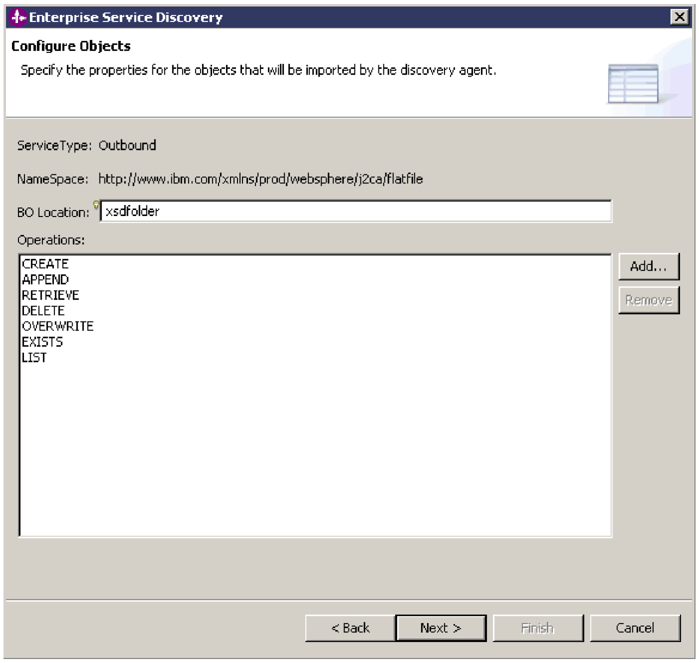
Use "Edit Query" to create a query and press "Run Query" to discover matching objects on the enterprise system.

The Find and Discover Enterprise Services window

9. In the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent. As illustrated in the graphic below, for the BO Location property, specify the folder name within the Business Integration module where the .xsd file should be generated. Click **Next**.

   The supported operations for Outbound include:

   - Create
   - Append
   - Retrieve
   - Delete
   - Overwrite
   - Exists
   - List



Enterprise Service Discovery

Configure Objects

Specify the properties for the objects that will be imported by the discovery agent.

ServiceType: Outbound

NameSpace: http://www.ibm.com/xmlns/prod/websphere/j2ca/flatfile

BO Location: xsdfolder

Operations:

CREATE
APPEND
RETRIEVE
DELETE
OVERWRITE
EXISTS
LIST

The Configure Objects window

10. In the Generate Artifacts window, do the following:
    a. Specify the module name where the SCA artifacts (business objects, their properties, import file, export file, and WSDL) should be saved, and then click **New** to create a new business integration module.

       The New Module window is displayed.
    b. In the New Module window, enter the Module name and click **Finish**.
    c. In the Generate Artifacts window, enter the folder name.
    d. In the Generate Artifacts window, select ″Use discovered connection properties″, enter the JNDI Lookup Name and click **Next**.
    e. In the Generate Artifacts window, specify the Flat Files ManagedConnectionFactory properties.

       Note: Properties marked with an asterisk (*) are required.
    f. In the Generate Artifacts window, specify the ResourceAdapter properties. You can also set logging and tracing files sizes and file name here as well.

       Note: Properties marked with an asterisk (*) are required.
    g. (Optional) Enter the BiDi properties.

       Note: Clear the**Turn BiDi Off** check box, otherwise BiDi support will not be invoked.
11. When you are finished entering property values, click **Finish**.

Now, you can use WebSphere Integration Developer to generate reference bindings for the SCA component that the enterprise service discovery wizard creates.

Related tasks

"Enabling logging" on page 42
The WebSphere Adapter for Flat Files maintains a log file that you can view to determine the status of event processing. All events and errors that relate to the adapter are tracked by the log file, along with the date, time, and event for each

log entry. Since the adapter logs an error message when it encounters an error or warning condition, the log file is a good source to start troubleshooting problems.

"Enabling tracing" on page 44
Tracing determines what level of errors or warnings are captured in the adapter log file. You can trace messages regarding adapter processing by defining a tracing level.

"Configuring properties" on page 34
You can configure adapter properties using the using WebSphere Process Server administrative console.

Related reference

"J2C connection factory properties" on page 36
This topic describes J2C connection factories and the properties used to configure a target enterprise information system (EIS) instance. These properties affect outbound processing and correspond to the ManagedConnectionFactory interface of the J2EE Connector Architecture Specification

"Resource adapter properties" on page 40
Using the enterprise service discovery wizard when you first configure the adapter (and later, via the WebSphere Process Server administrative console), you can configure Resource adapter properties. This category of properties includes logging and tracing properties and adapter-specific properties.

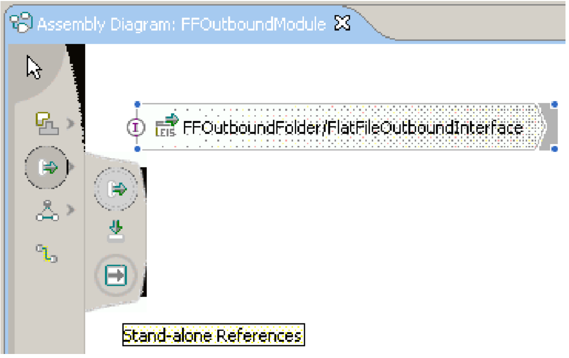
Generating reference bindings for outbound operations

After you have generated service artifacts, you can optionally generate reference bindings using WebSphere Integration Developer.

Prerequisite: An adapter project must be created and configured on your workspace. Once an adapter project is created, you must generate reference findings to bind to the service component.

Reference bindings are used by other WebSphere Business Integration SCA components to access the adapter. You create a reference to the adapter from the project module so as to link the adapter to the other server processes.

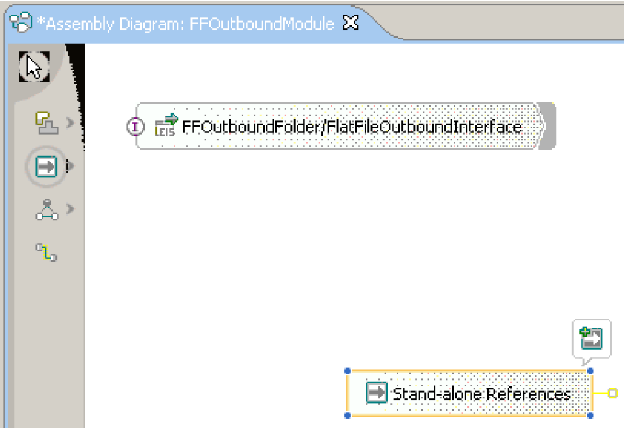
1. From the WebSphere Integration Developer window, switch to the Business Integration perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select Business Integration perspective in WebSphere Integration Developer.
2. In the Business Integration Perspective of WebSphere Integration Developer, right-click the module, and select **Open With** → **Assembly Editor**. The Assembly Diagram window appears with the modules Import component in view.
3. To create a Stand-alone Reference, click the top-most icon in the left-side (vertical) frame of the Assembly Diagram window.


Assembly Diagram: FFOutboundModule
FFOutboundFolder/FlatFileOutboundInterface
Stand-alone References

The Assembly Diagram window

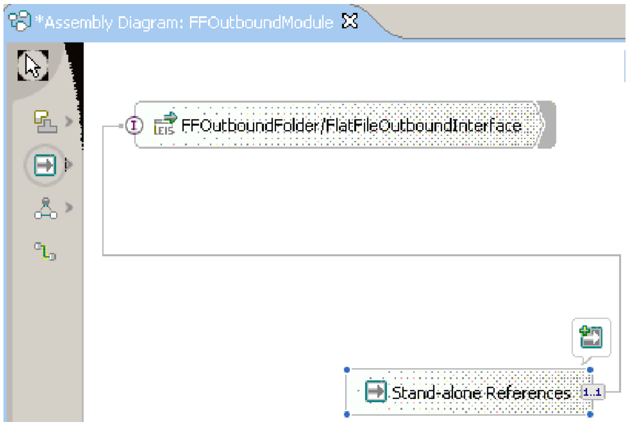
A new menu of icons appears. The graphic above illustrates the new menu of icons.

4. Create a Stand-alone Reference. As illustrated in the graphic below, drag the Stand-alone Reference icon into the Assembly Diagram window.


*Assembly Diagram: FFOutboundModule
FFOutboundFolder/FlatFileOutboundInterface
Stand-alone References

The Assembly Diagram window

5. Wire the Standalone Reference to the import module. As illustrated in the graphic below, click and drag the module's Import component to the new component. This draws a wire from the Import component to the new Stand-alone Reference.


*Assembly Diagram: FFOutboundModule
FFOutboundFolder/FlatFileOutboundInterface
Stand-alone References
1..1

The Assembly Diagram window

The Add Wire dialog box is displayed.

6. Click **OK**. The new Standalone Reference component displays in the Assembly Diagram window with a wire that connects it to the modules Import component.

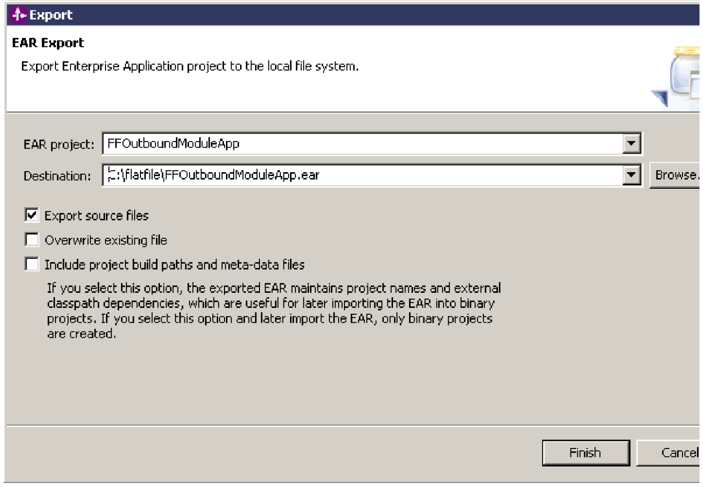
7. Under Dynamic web projects, right click the Web project and select **Properties**. The Web project contains the service client.
8. Select projects, the Flat Files RAR file, and select a default output folder.
9. Add your Web project as a dependent J2EE project with the Dependency editor.
   a. Open the Dependency editor by right clicking your module and selecting **Open Dependency Editor**.
   b. In the J2EE Project Selection window, select your project and click **OK**.
10. Save the file, **File** → **Save**.

Now, you are ready to export the EAR file.

Exporting the EAR file for outbound operations

Before you can run the project, you must export it to an EAR file using WebSphere Integration Developer.

1. From the WebSphere Integration Developer window, switch to the J2EE perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select J2EE perspective in WebSphere Integration Developer.
2. In the J2EE perspective window of WebSphere Integration Developer tool, right click the module and select Export from the pop-up menu. The Export window is displayed.
3. Select EAR file from the Export - Select window. The Export - EAR Export window is displayed.
4. As illustrated in the graphic below, in the EAR Export window, select the EAR project and the destination directory (the directory, including the EAR file name, where the project should be exported into).



Export

EAR Export

Export Enterprise Application project to the local file system.

EAR project: FFOutboundModuleApp

Destination: c:\flatfile\FFOutboundModuleApp.ear

☑ Export source files

☐ Overwrite existing file

☐ Include project build paths and meta-data files

If you select this option, the exported EAR maintains project names and external classpath dependencies, which are useful for later importing the EAR into binary projects. If you select this option and later import the EAR, only binary projects are created.

The EAR Export window

5. Click **Finish**.

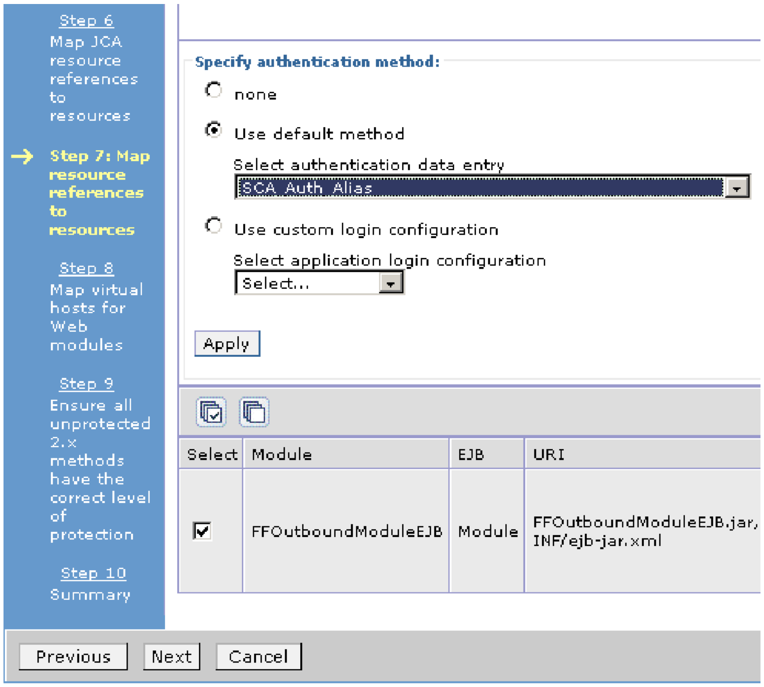
Now that you have exported the project to an Enterprise Application Archive (EAR) file, you are ready to install the application.

Installing the application for outbound operations

Installing the application project module is the last step of the deployment process.

When you install the application and run it, the adapter, which is embedded in the project module, runs as part of the installed application.

1. In the WebSphere Process Server administrative console, click **Applications** → **Install New Applications**.
2. Select the EAR file from the list of Enterprise Applications.
3. Under Path to the new application, specify the path of the EAR file, then click **Next**.
4. Continue to click **Next** until you reach Step 7: Map resource reference to resources.
5. As illustrated in the graphic below, under Specify authentication method:
   a. Select the "SCA Auth Alias" from the drop down menu.
   b. Click the check box for the module.
   c. Click **Apply**.



Step 6
Map JCA resource references to resources

→ **Step 7: Map resource references to resources**

Step 8
Map virtual hosts for Web modules

Step 9
Ensure all unprotected 2.x methods have the correct level of protection

Step 10
Summary

Specify authentication method:

○ none

● Use default method

Select authentication data entry
SCA Auth Alias

○ Use custom login configuration

Select application login configuration
Select...

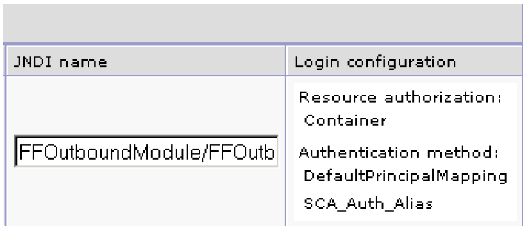
Apply

Select	Module	EJB	URI
☑	FFOutboundModuleEJB	Module	FFOutboundModuleEJB.jar, INF/ejb-jar.xml

Previous   Next   Cancel

The WebSphere Process Server administrative console

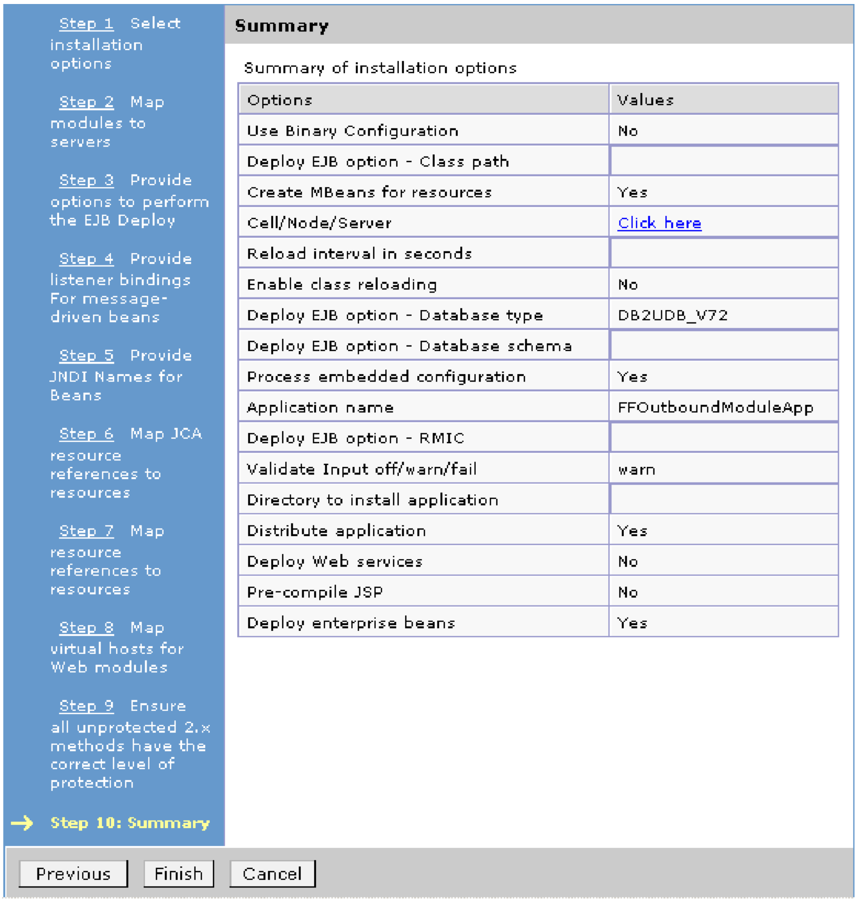
6. Scroll to the right to verify that the JNDI name matches the JNDI name present in the .xmi file of the EJB project.



JNDI name	Login configuration
FFOutboundModule/FFOutb	Resource authorization: Container Authentication method: DefaultPrincipalMapping SCA_Auth_Alias

The WebSphere Process Server administrative console

7. Click **Next**. A summary of all the installation options is displayed. Verify that all options are as you intended and click **Finish**.


Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide options to perform the EJB Deploy
Step 4 Provide listener bindings For message-driven beans
Step 5 Provide JNDI Names for Beans
Step 6 Map JCA resource references to resources
Step 7 Map resource references to resources
Step 8 Map virtual hosts for Web modules
Step 9 Ensure all unprotected 2.x methods have the correct level of protection
Step 10: Summary
Previous
Finish
Cancel

Summary

Summary of installation options

Options	Values
Use Binary Configuration	No
Deploy EJB option - Class path	
Create MBeans for resources	Yes
Cell/Node/Server	Click here
Reload interval in seconds	
Enable class reloading	No
Deploy EJB option - Database type	DB2UDB_V72
Deploy EJB option - Database schema	
Process embedded configuration	Yes
Application name	FFOutboundModuleApp
Deploy EJB option - RMIC	
Validate Input off/warn/fail	warn
Directory to install application	
Distribute application	Yes
Deploy Web services	No
Pre-compile JSP	No
Deploy enterprise beans	Yes

The WebSphere Process Server Administrative Console

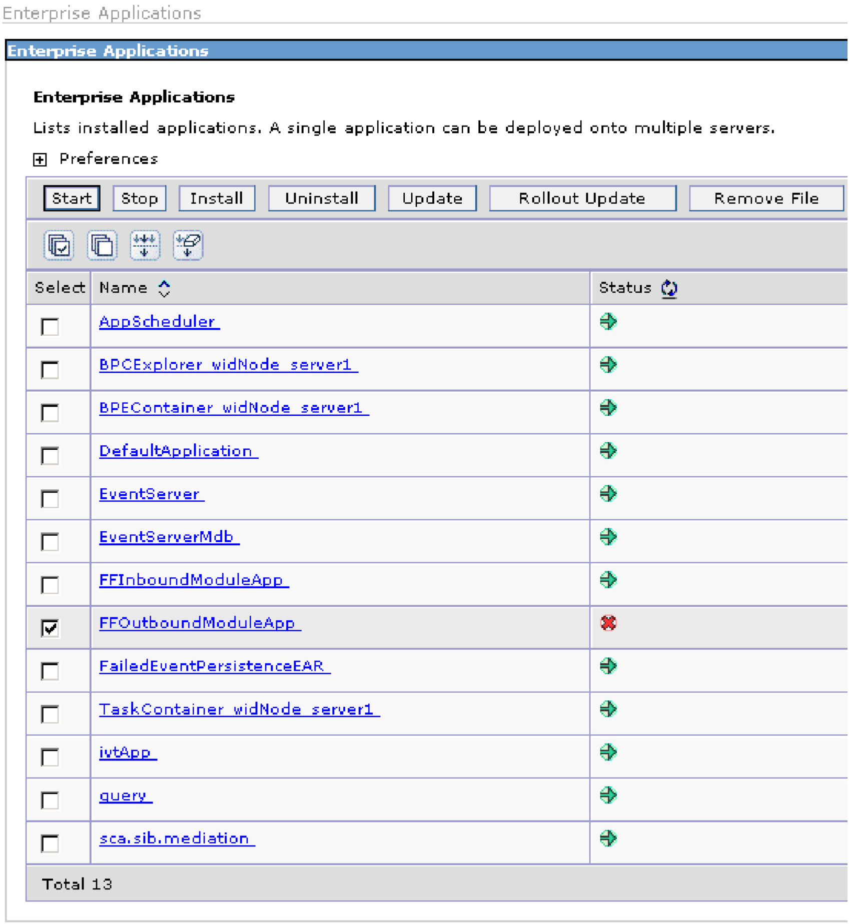
8. A list of installation messages appears. Confirm that the message Application installed successfully is included at the end of the list.
9. Click the **Save to Master Configuration** link that appears at he end of the list of installation messages. The Enterprise Applications window is displayed.
10. Click **Save** to save the application. The application is now deployed and the Enterprise Applications window for the deployed application is displayed.
11. If necessary edit the J2C activation specification (ActivationSpecification) properties.

The application is now deployed and properly configured. The next step is to run the application.

Running the application for outbound operations

After you have deployed the application, you can run it. Since the adapter is embedded in the application project, when you run the application, the adapter is triggered to start running.

1. In the WebSphere Process Server administrative console, click **Applications** → **Enterprise Applications**.
2. As the graphic below illustrates, select the check box for the application and click the **Start** button. The application starts running.



Enterprise Applications

Lists installed applications. A single application can be deployed onto multiple servers.

⊞ Preferences

Select	Name	Status
☐	AppScheduler	➔
☐	BPCExplorer_widNode_server1	➔
☐	BPEContainer_widNode_server1	➔
☐	DefaultApplication	➔
☐	EventServer	➔
☐	EventServerMdb	➔
☐	FFInboundModuleApp	➔
☑	FFOutboundModuleApp	✖
☐	FailedEventPersistenceEAR	➔
☐	TaskContainer_widNode_server1	➔
☐	ivtApp	➔
☐	query	➔
☐	sca.sib.mediation	➔

Total 13

The WebSphere Process Server administrative console

Configuring the adapter

This topic introduces the task of configuring the adapter, which is done by setting configuration properties.

Configuring properties

You can configure adapter properties using the using WebSphere Process Server administrative console.

The adapter has four categories of properties that are configured using WebSphere Process Server administrative console:

- J2C connection factory properties (which correspond to the ManagedConnectionFactory interface)
- J2C activation specification properties (which correspond to the ActivationSpecification interface)
- Custom properties (also referred to as "default configuration properties")
- Resource adapter properties

Note: You can also configure properties for the Flat Files adapter through the enterprise service discovery wizard during the generation of service artifacts.

To configure properties using the WebSphere Process Server administrative console, follow these steps.

1. Start the console.
2. Under Resources, select **Resource Adapters**.
3. Under Resource Adapters, select WebSphere Adapter for Flat Files. The General Properties page appears.
4. Under Additional Properties, select one of the following categories of properties that you would like to change:
   - J2C connection factories, to configure the ManagedConnectionSpec properties, which are used to configure a target enterprise information system instance.
   - J2C Activation specifications, to configure message endpoint properties.
   - Custom properties, to configure default configuration properties that are shared by all WebSphere adapters.
5. Do one of the following:
   - If you selected J2C connection factories, select the name of the J2C connection factory you want to configure and then select Connection pool properties, Advanced connection factory properties, or Custom properties, depending on which J2C connection factory properties you want to configure. Custom properties are those J2C connection factory properties that are unique to the WebSphere Adapter for Flat Files. Connection pool and Advanced connection factory properties are properties you configure if you are developing your own adapter.
   - If you selected J2C Activation specifications, select the name of the J2C activation specification that you want to configure. Then select the name of the message endpoint property you want to configure and set the value as desired.
   - If you selected Custom properties, the Custom properties page appears. Select the name of the default configuration property you wish to configure and set the value as desired.

Related tasks

"Generating service artifacts for inbound operations" on page 14
The generation of service artifacts for inbound operations is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Configuring a service for outbound operations" on page 25
The configuration process is done using the enterprise service discovery wizard in WebSphere Integration Developer.

Configuration properties for the WebSphere Adapter for Flat Files

This topic provides a list of the configuration properties of WebSphere Adapter for Flat Files.

WebSphere Adapter for Flat Files has several categories of configuration properties.

The following table describes the categories of adapter configuration properties.

Configuration property category	Description
J2C connection factory	Used to configure outbound processing
J2C activation specification	Used to configure inbound processing

Configuration property category	Description
Resource adapter properties	Used to configure features such as logging and tracing
Enterprise service discovery connection properties	Used during initial adapter deployment to configure either inbound or outbound processing and bidirectional enablement.

J2C connection factory properties

This topic describes J2C connection factories and the properties used to configure a target enterprise information system (EIS) instance. These properties affect outbound processing and correspond to the ManagedConnectionFactory interface of the J2EE Connector Architecture Specification

A J2C connection factory manages connection pooling. It provides configuration information for outbound operations via the resource adapter.

The ″ManagedConnectionFactory configuration properties″ table defines the configuration properties that pertain to a J2C connection factory.

ManagedConnectionFactory configuration properties

Property name	Type	Globalized	Bidirectional transformation support (Yes/No)	Description
OutputDirectory	String	Yes	Yes	Output directory for outbound operations.
OutputFileName	String	Yes	Yes	Output file name for outbound operations.

Related tasks

"Configuring a service for outbound operations" on page 25
The configuration process is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Deploying and configuring outbound operations for scenario 2" on page 53
Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for outbound processing.

J2C activation specification properties

This topic describes J2C activation specification properties (also referred to as message end point properties), which correspond to the ActivationSpecification interface of the J2EE Connector Architecture Specification.

The ActivationSpecification properties hold the inbound event processing configuration information for a message endpoint.

The ″ActivationSpecification properties″ table defines the configuration properties that pertain to the ActivationSpec interface.

ActivationSpecification properties

Property name	Type	Globalized	Bidirectional transformation support (Yes/No)	Description
ArchiveDirectory	String	Yes	Yes	Directory where the processed event files would be archived by the adapter. Required value: Yes, if ArchivingProcessed =true
ArchivingProcessed	Boolean	Yes	No	Boolean property that determines if the adapter should archive the processed events. Default value: True Required value: Yes
EventDirectory	String	Yes	Yes	Directory where the event files get stored by the backend EIS. Required value: Yes
EventFileMask	String	Yes	Yes	Specifies the filter for the event files. The file filter matches patterns only based on the wildcard *. Default value: *.* Required value: Yes
FFDatabaseName	String	Yes	Yes	Name of the logical database in which FFEventTable resides. This database is located in the Cloudscape DBMS that is contained within the WebSphere Process Server. Default value: FFDB Required value: Yes
FFEventTableName	String	Yes	Yes	Name of the table that will be utilized by the adapter for tracking event status values. Default value: FFLOG Required value: Yes
FileChunkSize	Integer	Yes	No	Specifies the size of each chunk in bytes, when the file needs to be split to chunks. When the property FileSplitThreshold is -1, this property remains inactive. Default value: 8000

Property name	Type	Globalized	Bidirectional transformation support (Yes/No)	Description
FileSplifThreshold	Integer	Yes	No	Specifies the threshold file size for splitting files in bytes. All files above this file size would be split into chunks. When the attribute is blank, splitting is not enabled.
SortEventFiles	String	Yes	No	Determines the sorting order of event files being polled. The values supported include: Filename - sort ascending on filename Timestamp - sort ascending on last modified timestamp<Blank> - not sorted To support globalization, the sorting of filenames is provided according to the system locale. The ICU4J package will be used to track the locales and the rules corresponding to the locales. Default value: <blank> (= not sorted)

Related tasks

"Generating service artifacts for inbound operations" on page 14
The generation of service artifacts for inbound operations is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Deploying and configuring inbound operations for scenario 2" on page 50
Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for inbound processing.

Custom properties

This topic describes the default configuration properties for WebSphere Adapter for Flat Files.

The ″Custom properties″ table describes the default configuration properties.

Custom properties

Property name	Type	Globalized	Bidirectional transformation support (Yes/No)	Description
AutocreateEDT	Boolean	Yes	No	When the value is set to true, an in-memory table gets created and event management framework uses the same to track event delivery. Required: Yes Default value: True

Property name	Type	Globalized	Bidirectional transformation support (Yes/No)	Description
DeliveryType	String	Yes	No	Specifies the type of delivery to the endpoint as ordered or random. Required: Yes Default value: ORDERED
EDTDatabaseName	String	Yes	Yes	Name of the database that will be utilized by event management framework.
EDTDriverName	String	Yes	No	Name of the driver to the event data table database.
EDTTableName	String	Yes	Yes	Name of the table on the event data table database that will be utilized by the event management framework.
EDTUserName	String	Yes	Yes	User ID credential to access the event data table database. This property is enabled for use with bidirectional languages.
EDTUserPassword	String	Yes	Yes	Password credential to access the event data table database.
PollPeriod	Integer	Yes		The interval in milliseconds when the event directory will be polled for events. Default value: 2000
PollQuantity	Integer	Yes	No	The number of event files that would be picked up for processing during each poll. Default value: 10

Related tasks

"Generating service artifacts for inbound operations" on page 14
The generation of service artifacts for inbound operations is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Deploying and configuring inbound operations for scenario 2" on page 50
Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for inbound processing.

Enterprise service discovery connection properties

Connection properties include the following categories of properties that you configure when you initially deploy the adapter via the enterprise service discovery wizard: outbound and inbound connection properties required for performing metadata discovery and for bidirectional configuration properties.

The inbound connection description is written to the Export file.

Inbound connections

Property	Description	Elements in the file
ActivationSpec	The instance of class representing the Flat Files resource adapter activation spec.	type in the connection element
Resource Adapter	The instance of class representing the Flat Files resource adapter.	resourceAdapter element

Outbound connections

Property	Description	Elements in the file
Managed Connection Factory	The instance of class representing the Flat Files resource adapter activation spec.	type in the connection element
Resource Adapter	The instance of class representing the Flat Files resource adapter.	resourceAdapter element

Resource adapter properties

Using the enterprise service discovery wizard when you first configure the adapter (and later, via the WebSphere Process Server administrative console), you can configure Resource adapter properties. This category of properties includes logging and tracing properties and adapter-specific properties.

Logging and tracing properties

The following table describes the logging and tracing properties for the Flat Files adapter.

Property name	Type	Globalized	Description
LogFileSize	Integer	Yes	Size of the log files in kilobytes. If no values is specified the file will have no maximum size.
LogFileName	String	Yes	The full path of the log file.
LogNumberOfFiles	Integer	Yes	The number of log files to use. When a log file reaches its maximum size it will start using another log file. If no values is specified it will be set to 1.
TraceFileSizeMax	Integer	Yes	Size of the trace files in kilobytes. If no values is specified the file will have no maximum size.
TraceFileName	String	Yes	The full path to the trace file.
TraceNumberOfFiles	Integer	Yes	The number of trace files to use. When a trace file reaches its maximum size it will start using another trace file. If no values is specified it will be set to 1.

Related tasks

"Generating service artifacts for inbound operations" on page 14
The generation of service artifacts for inbound operations is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Configuring a service for outbound operations" on page 25
The configuration process is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Deploying and configuring inbound operations for scenario 2" on page 50
Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for inbound processing.

"Deploying and configuring outbound operations for scenario 2" on page 53
Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for outbound processing.

Troubleshooting and support

The troubleshooting and support documentation helps you find the cause of problems or errors with the WebSphere Adapter for Flat Files. Support and contact information is provided to help you report and resolve problems.

The following topics provide information about troubleshooting and support for the adapter.

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli®, Lotus®, and Rational® products, as well as DB2® and WebSphere products that run on Windows or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:
  - **Online:** Go to the Passport Advantage Web page and click How to Enroll.
  - **By phone:** For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries®, pSeries®, and iSeries™ environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

- Determine the business impact of your problem.
- Describe your problem and gather background information.
- Submit your problem to IBM Software Support.

1. Determine the business impact of your problem. When you report a problem to IBM, you will be asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

Severity	Description
Severity 1	Critical business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.
Severity 2	Significant business impact: The program is usable but is severely limited.
Severity 3	Some business impact: The program is usable with less significant features (not critical to operations) unavailable.
Severity 4	Minimal business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.

2. Describe your problem and gather background information. When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:
   - What software versions were you running when the problem occurred?
   - Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
   - Can the problem be recreated? If so, what steps led to the failure?
   - Have any changes been made to the system? (For example, hardware, operating system, networking software, and so on.)
   - Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.
3. Submit your problem to IBM Software Support. You can submit your problem in one of two ways:
   - **Online:** Go to the Submit and track problems page on the IBM Software Support site. Enter your information into the appropriate problem submission tool.
   - **By phone:** For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support will create an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail.

Whenever possible, IBM Software Support will provide a workaround for you to implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM product support Web pages daily, so that other users who experience the same problem can benefit from the same resolutions.

Enabling logging

The WebSphere Adapter for Flat Files maintains a log file that you can view to determine the status of event processing. All events and errors that relate to the adapter are tracked by the log file, along with the date, time, and event for each log entry. Since the adapter logs an error message when it encounters an error or warning condition, the log file is a good source to start troubleshooting problems.

Logging for the Flat Files Adapter is enabled through the WebSphere Process Server administrative console. Follow the steps below to enable logging.

1. Start the WebSphere Process Server administrative console.
2. From the administrative console, select **Troubleshooting** → **Logs and Trace**.
3. Click the server link that corresponds to your server.
4. Click ″Change Log Detail levels″.
5. Click **Component** to specify a log detail level for individual components or click **Groups** to specify a log detail for a predefined group of components.
6. Select the logging level that you need. The "Logging levels" table describes the different logging levels that can be set through the WebSphere Process Server administrative console.

   Note: To view log events that are below the Detail Level, you must enable the Diagnostic Trace Service. Log events that are at Detail Level or above can be viewed in the SystemOut log, the IBM Service log (when enabled), or the Diagnostic Trace Service (when enabled).

   Logging levels

Level	Indicator	Description
Audit	A	Significant event affecting server state or resources
Config	C	Configuration change or status.
Detail	D	General information detailing subtask progress.
Fatal	F	Task cannot continue. Component cannot function.
Info	I	General information outlining overall task progress
Severe	E	Task cannot continue. Component can still function. This also includes conditions that indicate an impending fatal error - i.e. reporting on situations that strongly suggest that resources are on the verge of being depleted.
Warning	W	Potential error or impending error. This also includes conditions that indicate a progressive failure - for example, the potential leaking of resources.

7. Click **Apply** to save your changes.

   Related tasks

   "Generating service artifacts for inbound operations" on page 14
   The generation of service artifacts for inbound operations is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Configuring a service for outbound operations" on page 25
The configuration process is done using the enterprise service discovery wizard in WebSphere Integration Developer.

"Deploying and configuring inbound operations for scenario 2" on page 50
Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for inbound processing.

"Deploying and configuring outbound operations for scenario 2" on page 53
Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for outbound processing.

Enabling tracing

Tracing determines what level of errors or warnings are captured in the adapter log file. You can trace messages regarding adapter processing by defining a tracing level.

The trace levels can be configured in the WebSphere Process Server administrative console. Follow the steps below to enable and set tracing levels.

1. Start the WebSphere Process Server administrative console.
2. From the administrative console, select **Troubleshooting** → **Logs and Trace**.
3. Click the server link that corresponds to your server.
4. Click ″Change Log Detail levels″.
5. Click **Component** to specify a log detail level for individual components or click **Groups** to specify a log detail for a predefined group of components.
6. Select the Tracing level that you need. The ″Tracing levels″ table describes the different tracing levels that can be set through the WebSphere Process Server administrative console.

   Tracing levels

Level	Indicator	Description
Fine	1	General trace. Includes broad actions being taken by adapter such as establishing a connection to the enterprise information system, converting an event in the enterprise information system to a business object (only key values), processing a business object (only key values).
Finer	2	Detailed trace that provides more granular information on the logic being performed by the adapter including the various application-specific information calls being made to the enterprise information system and any parameters or return values.

Level	Indicator	Description
Finest	3	This is the most detailed level and should include method entry / exit / return values. Complete business object dumps should be included. At this level, all detail needed to debug problems should be provided.

7. Click **Apply** to save your changes.

   Related tasks

   "Generating service artifacts for inbound operations" on page 14
   The generation of service artifacts for inbound operations is done using the enterprise service discovery wizard in WebSphere Integration Developer.

   "Configuring a service for outbound operations" on page 25
   The configuration process is done using the enterprise service discovery wizard in WebSphere Integration Developer.

   "Deploying and configuring inbound operations for scenario 2" on page 50
   Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for inbound processing.

   "Deploying and configuring outbound operations for scenario 2" on page 53
   Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for outbound processing.

Enabling the Common Event Infrastructure (CEI)

This topic describes how to enable the Common Event Infrastructure (CEI) for the adapter.

You must publish the IBMWebSphere

Adapters Event Definitions file to the CEI catalog before you can set these event definitions. For instruction on how to do this, refer to the CEI documentation found on the WebSphere Process Server web site at http://www.ibm.com/software/integration/wps.

1. Start the WebSphere administrative console.
2. Go to **Troubleshooting** → **Log and Trace** and select <your server name>.
3. There are many options for the General Properties. Select **Change Log Detail Level**, and then select **com.ibm.j2ca.*** for JCA components. Under this section there is a subcomponent for each adapter type:
   - com.ibm.j2ca.flatfile.* (WebSphere Adapter for Flat Files)
   - com.ibm.j2ca.jdbc.* (WebSphere Adapter for JDBC)
   - com.ibm.j2ca.peoplesoft.* (WebSphere Adapter for PeopleSoft)
   - com.ibm.j2ca.sap.* (WebSphere Adapter for SAP)
   - com.ibm.j2ca.siebel.* (WebSphere Adapter for Siebel)
4. Select the component that matches your adapter. Each adapter component has two subcomponents, one for logging and one for CEI. They are:
   - *subcomponent name*.log.*adapter id*

- *subcomponent name*.cei.*adapter id*

   For example, com.ibm.j2ca.siebel.cei.<AdapterID1>. For each instance of a deployed adapter, the system will show a separate ID.

5. Select the CEI adapter ID that you want to enable.
6. From the drop-down menu, you can choose from the following:
   - off - turn CEI off
   - fine- turn CEI on with Event Content set to Empty
   - finer- turn CEI on with Event Content set to Digest
   - finest- turn CEI on with Event Content set to Full
   - all - same as finest

For information on what each Event Content level means (Empty, Digest and Full), and for more information on using the Common Base Event model and the Common Event Infrastructure, refer to the documentation on the WebSphere Process Server web site at http://www.ibm.com/software/integration/wps

Using the sample application

The adapter provides a sample application that illustrates how you deploy an application package and how the adapter processes business objects. The application presents two scenarios, one for each audience of the adapter. The adapter audience consists of two users: the application integrator and the data integrator.

For each of the samples provided, the two scenarios presented are as follows:

Sample application scenarios

Scenario	Description	Audience
Scenario 1	• Provides the already-generated artifacts and illustrates how the adapter processes business objects. Using the enterprise service discovery wizard to generate artifacts is not required in this scenario. • Targeted at an audience that is responsible for assembling application components into a solution and preparing this solution for testing and deployment.	Application integrator

Scenario	Description	Audience
Scenario 2	• Illustrates how you use the enterprise service discovery wizard to discover application components and develop the business objects that the adapter processes. • Targeted at an audience with the same responsibilities as the application integrator, but is further responsible for enabling access to a range of data sources for the application developers.	Data integrator

Structure of the application package

The sample application files are installed when you install the adapter. They are packaged in an archive file that is installed in the samples folder.

Scenario 1: All-inclusive set of sample files

For Scenario 1, targeted at the Application Integrator, the sample application package includes all the required artifacts, so you do not need to use the enterprise service discovery wizard to obtain these. In a real-world deployment of the adapter, you would generate all these artifacts and configure the adapter using the enterprise service discovery wizard, as illustrated in Scenario 2.

The files for Scenario 1 are archived in the following file: \adapter\flatfile\samples\nonemdsamples.zip.

1. Navigate to the samples folder and uncompress the nonemdsamples.zip. This extracts the FlatFileInboundModuleApp.ear and the FlatFileOutboundModuleApp.ear.
2. The FlatFileInboundModuleApp.ear includes the following files:
   - FlatFileInboundModule.jar
   - FlatFileInboundModuleEJB.jar
   - FlatFileInboundModuleEJBClient.jar
   - FlatFileInboundModuleWeb.war
   - CWYFF_FlatFile.rar
   - application.xml
   - deployment.xml
3. The FlatFileOutboundModuleApp.ear includes the following files:
   - FlatFileOutboundModule.jar
   - FlatFileOutboundModuleEJB.jar
   - FlatFileOutboundModuleEJBClient.jar
   - FlatFileOutboundModuleWeb.war
   - CWYFF_FlatFile.rar
   - applicaiton.xml
   - deployment.xml

- MyWebProject.war

Scenario 2: enterprise service discovery wizard requirement

For Scenario 2, targeted at the Data Integrator, the sample application package provides the capability for you to create the SCA artifacts and configure the adapter using the enterprise service discovery wizard. You can, however, access copies all the files that you eventually generate using the enterprise service discovery wizard, as a way of verifying that what you create with the enterprise service discovery wizard is indeed correct and what the application expects in order to run properly.

All the necessary files for scenario 2 are in \adapter\flatfile\samples\emdsamples.zip. Copies of those you will generate with enterprise service discovery wizard can be found in \adapter\flatfile\samples\nonemdsamples.zip.

1. Navigate to the samples folder and uncompress the emdsamples.zip file. This extracts the CWYFF_FlatFile.rar file.
2. The enterprise service discovery wizard will import this RAR file and generate the service artifacts as per the your specified configuration.

Business object structure

The Business object definition for Flat Files Resource Adapter contains the required information for executing the operations of the Flat Files adapter and for Inbound operations.

The Flat Files adapter has the same business object structure for both inbound and outbound operations. The business object schema consists of the following attributes:

- directoryPath
- fileName
- inputBytes
- outputBytes
- outputString

The business graph that contains the business object structure is the FlatFileBG.xsd file. This consists of an element of complex type FlatFile.

Deploying and configuring inbound operations for scenario 1

Scenario 1 of the sample application provides a configured instance of the adapter and all the necessary SCA artifacts. Simply import the EAR file to your WebSphere Integration Developer workspace using WebSphere Integration Developer and use the WebSphere Business Integration Component Test Tool to complete the configuration process.

1. Since this Scenario provides an instance of the adapter that is already configured, you simply import the EAR file in your project using WebSphere Integration Developer. In WebSphere Integration Developer, select **File** → **Import** and select the CWYFF_FlatFile.rar.
2. Switch to the Business Integration perspective and create a new module, FlatFileInboundModule. To create the new module, right click the Business Integration panel and select **New** → **Module** from the pop-up menu.
3. Extract the contents of the FlatFileInboundModule.jar.

4. In WebSphere Integration Developer, refresh the FlatFileModule in the Business Integration perspective. Right click the FlatFileInboundModule and select refresh from the pop-up menu.
5. Add the CWYFF_FlatFile.rar as a dependent J2EE project. Right click the FlatFileInboundModule in the Business Integration perspective and select **Open Dependency Editor**.
6. Edit the ActivationSpecification properties. In WebSphere Integration Developer, double click the first element under FlatFileInboundModule to open the Assembly Editor.
   a. Select **Properties** → **Binding** → **Connection** → **ActivationSpec Properties**
   b. Edit the ActivationSpec properties according to your specific system requirements.
   c. Click **Save** when you are finished editing the ActivationSpec properties.
7. In the Business Integration perspective, right click FlatFileInboundModule.
8. Select **Test** → **Test Module** from the pop-up menu. The Events window is displayed.
9. In the Events window, place a sample event file in the event directory.
   a. Verify the event file has an ″.in″ extension.
   b. To start the WebSphere Process Server and install the inbound application, click **Continue**.

Deploying and configuring outbound operations for scenario 1

Scenario 1 of the sample application provides a configured instance of the adapter and all the necessary SCA artifacts. Simply import the EAR file to your WebSphere Integration Developer workspace using WebSphere Integration Developer and use the WebSphere Business Integration Component Test Tool to complete the configuration process.

1. Since this Scenario provides an instance of the adapter that is already configured, you simply import the EAR file in your project using WebSphere Integration Developer. In WebSphere Integration Developer, select **File** → **Import** and select the CWYFF_FlatFile.rar.
2. Switch to the Business Integration perspective and create a new module, FlatFileOutboundModule. To create the new module, right click the Business Integration panel and select **New** → **Module** from the pop-up menu.
3. Extract the contents of the FlatFileOutboundModule.jar.
4. In WebSphere Integration Developer, refresh the FlatFileModule in the Business Integration perspective. Right click the FlatFileOutboundModule and select refresh from the pop-up menu.
5. Add the CWYFF_FlatFile.rar as a dependent J2EE project. Right click the FlatFileInboundModule in the Business Integration perspective and select **Open Dependency Editor**.
6. Edit the ManagedConnectionFactory properties. In WebSphere Integration Developer, double click the first element under FlatFileOutboundModule to open the Assembly Editor.
   a. Select **Properties** → **Binding** → **Connection** → **ManagedConnectionFactory properties**
   b. Edit the ManagedConnectionFactory properties according to your specific system requirements.
   c. Click **Save** when you are finished editing the ActivationSpecification properties.

7. In the Business Integration perspective, right click FlatFileOutboundModule.
8. Select **Test** → **Test Module** from the pop-up menu. The Events window is displayed.
9. In the Events window, select an operation.
   a. Populate the values in the business object based on the operation that you want to perform.
   b. To start the WebSphere Process Server and install the outbound application, click **Continue**.

Deploying and configuring inbound operations for scenario 2

Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for inbound processing.

Before you begin to deploy and configure, import the CWYFF_FlatFile.RAR file into the project. For this task, you must run the enterprise service discovery within WebSphere

Integration Developer to set adapter configuration properties. To change the property values later, use the WebSphere Process Server administrative console.

1. From the WebSphere Integration Developer window, switch to the Business Integration perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select Business Integration perspective in WebSphere Integration Developer.
2. In the Business Integration Perspective of WebSphere Integration Developer, right-click the frame and from the pop-up menu, select **New** → **Enterprise Service Discovery**. The Select an Enterprise Service Resource Adapter window is displayed.
3. Select the **IBM WebSphere Adapter for Flat Files (version 6.0.0) from the 'CWYFF_FlatFile' Connector Project** and click **Next**. The Configure Settings for Discovery Agent window is displayed.
4. In the Configure Settings for Discovery Agent window, set the connection properties by selecting Inbound for the ServiceType.
5. Optional. At the bottom of the Configure Settings for Discovery Agent window, click the **Show Advanced** button. You can set logging and tracing levels here. Click **Next**.
6. In the Find and Discover Enterprise Services window, click the **Run Query** button to display the meta-data tree of the Flat Files adapter.
7. Under Objects discovered by query, for inbound operations, select the Inbound root node of the meta-data tree. The objects that can be selected for import are displayed in the Objects discovered by query box. Highlight the object you would like to import, then click the **Add Selected** button to add the objects to the Objects to be imported box. Click **Next**.

   Note: To remove objects from the Objects Discovered by query text box, highlight the object that you would like to remove and click the **Remove Selected** button.
8. In the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent. For the BO Location property, specify the folder name within the Business Integration module where the .xsd file should be generated. Click **Next**.

   The supported operations for Inbound include:

- Read

9. In the Generate Artifacts window, do the following:
   a. Click **New** to create a new business integration module.
   b. In the New Module window, enter the Module name, FlatFileInboundModule, and click **Finish**.
   c. In the Generate Artifacts window, enter the folder name, FlatFileInboundFolder.
   d. Under ″Specify the connection properties which will be used to connect to the Enterprise Information System at runtime″, select **Use discovered connection properties**.
   e. Enter the Event Distribution Database properties.

      Note: Required properties are indicated with a * next to the property box.
   f. Enter the Flat Files adapter ActivationSpecification properties.

      Note: Properties marked with an asterisk (*) are required.
   g. Enter the logging and tracing values.

      Note: Required properties are indicated with a * next to the property box.
10. When you are finished entering property values, click **Finish**. The new module is added to the Business Integration Perspective, along with all its artifacts.

    Now, you can use WebSphere Integration Developer to generate reference bindings for the SCA component that the enterprise service discovery tool creates. Reference bindings are used by other WebSphere Business Integration SCA components to access the adapter. You create a reference to the adapter from the project module so as to link the adapter to the other server processes.
11. In the Business Integration Perspective of WebSphere Integration Developer, right-click the module, and select **Open With → Assembly Editor**. The Assembly Diagram window appears with the modules Import component in view.
12. To create a new component, click the top-most icon in the left-side (vertical) frame of the Assembly Diagram Window. A new menu of icons appears.
13. Create a Component with no implementation. Drag the Create a Component icon into the Assembly Diagram window.
14. Create wiring from the FlatFileInboundInterface to this new component. Click and drag the module‘s Import component to the new component. This draws a wire from the Import component to the new component. The Add Wire dialog box is displayed.
15. Click **OK**. The new Standalone Reference component displays in the Assembly Diagram window with a wire that connects it to the modules Import component.
16. Create a Java component that will acts like an endpoint listener. During event delivery of the inbound operation, the adapter invokes the READ method of the component implementation and passes the inbound business object as a parameter.
    a. Right click the component and select **Generate implementation → Java**.
    b. Select the package where the Java code should be created and click **OK**.
    c. Enter the Package name, listener, when prompted and click **OK**.
    d. Click **OK** in the Generate Implementation window.

   e. In the generated Java implementation, scroll to the READ method. You can add custom code inside the READ method to process the delivered business object according to your specific business needs. For example, you can add print statements to detect event delivery.
   f. Save the file, **File** → **Save**.
17. Edit the ActivationSpecificaiton properties. In WebSphere Integration Developer, double click the first element under FlatFileInboundModule to open the Assembly Editor.
   a. Select **Properties** → **Binding** → **Connection** → **ActivationSpec Properties**
   b. Edit the ActivationSpecification properties according to your specific system requirements.
   c. Click **Save** when you are finished editing the ActivationSpecification properties.
18. In the Business Integration perspective, right click FlatFileInboundModule.
19. Select **Test** → **Test Module** from the pop-up menu. The Events window is displayed.
20. In the Events window, place a sample event file in the event directory.
   a. Verify the event file has an ″.in″ extension.
   b. To start the WebSphere Process Server and install the inbound application, click **Continue**.

The adapter application package, including all its SCA artifacts, is now configured and deployed. You can now run the sample application.

Related tasks

"Enabling logging" on page 42
The WebSphere Adapter for Flat Files maintains a log file that you can view to determine the status of event processing. All events and errors that relate to the adapter are tracked by the log file, along with the date, time, and event for each log entry. Since the adapter logs an error message when it encounters an error or warning condition, the log file is a good source to start troubleshooting problems.

"Enabling tracing" on page 44
Tracing determines what level of errors or warnings are captured in the adapter log file. You can trace messages regarding adapter processing by defining a tracing level.

Related reference

"Custom properties" on page 38
This topic describes the default configuration properties for WebSphere Adapter for Flat Files.

"J2C activation specification properties" on page 36
This topic describes J2C activation specification properties (also referred to as message end point properties), which correspond to the ActivationSpecification interface of the J2EE Connector Architecture Specification.

"Resource adapter properties" on page 40
Using the enterprise service discovery wizard when you first configure the adapter (and later, via the WebSphere Process Server administrative console), you can configure Resource adapter properties. This category of properties includes logging and tracing properties and adapter-specific properties.

Deploying and configuring outbound operations for scenario 2

Scenario 2 of the sample application requires you to use the enterprise service discovery wizard to deploy the application package, configure the adapter, and generate the SCA artifacts for outbound processing.

Before you begin to deploy and configure, import the CWYFF_FlatFile.RAR file into the project. For this task, you must run the enterprise service discovery within the IBMWebSphere

Integration Developer to set adapter configuration properties. To change the property values later, use the WebSphere Process Server administrative console.

1. From WebSphere Integration Developer window, switch to the Business Integration perspective by selecting **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed. Select Business Integration perspective in WebSphere Integration Developer.
2. In the Business Integration Perspective of WebSphere Integration Developer, right-click the frame and from the pop-up menu, select**New** → **Enterprise Service Discovery**. The Select an Enterprise Service Resource Adapter window is displayed.
3. Select the **IBM WebSphere Adapter for Flat Files (version 6.0.0) from the ’CWYFF_FlatFile’ Connector Project** and click **Next**. The Configure Settings for Discovery Agent window is displayed.
4. In the Configure Settings for Discovery Agent window, set the connection properties by selecting Outbound for the ServiceType. Outbound is the default ServiceType.
5. At the bottom of the Discovery Agent Initialize Properties window, click the Show Advanced button. You can set logging and tracing levels here. Click **Next**.
6. In the Find and Discover Enterprise Services window, click **Run Query**.
7. Under Objects Discovered, select the objects that you want to import. The objects that can be selected for import are displayed in the Objects Discovered box. Highlight the object you would like to import, then click the **Add Selected** button to add the objects to the Objects to be imported box. Click **Next**.

   Note: To remove objects from the Objects Discovered box, highlight the object that you would like to remove and click the **Remove Selected** button.
8. In the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent. For the BO Location property, specify the folder name within the Business Integration module where the .xsd file should be generated. Click **Next**.

   The supported operations for Outbound include:
   - Create
   - Append
   - Retrieve
   - Delete
   - Overwrite
   - Exists
   - List
9. In the Generate Artifacts window, do the following:

a. In the Generate Artifacts window, specify the module name where the SCA artifacts (business objects, their properties, import file, export file, and WSDL) should be saved, and then click **New** to create a new business integration module. The New Module window is displayed.

b. In the New Module window, enter the Module name and click **Finish**. The module name can be FlatFileOutboundModule.

c. In the Generate Artifacts window, enter the folder name. The folder name can be FlatFileOutboundFolder.

d. Under ″Specify the connection properties which will be used to connect to the Enterprise Information System at runtime″, select **Use discovered connection properties**.

e. In the Generate Artifacts window, specify the Flat Files ManagedConnectionFactory properties.

   Note: Properties marked with an asterisk (*) are required.

f. In the Generate Artifacts window, specify the ResourceAdapter properties. You can also set logging and tracing files sizes and file name here as well.

   Note: Properties marked with an asterisk (*) are required.

10. When you are finished entering property values, click **Finish**. The new module is added to the Business Integration Perspective, along with all its artifacts.

    Now, you can use WebSphere Integration Developer to generate reference bindings for the SCA component that the enterprise service discovery tool creates. Reference bindings are used by other WebSphere Business Integration SCA components to access the adapter. You create a reference to the adapter from the project module so as to link the adapter to the other server processes.

11. In the Business Integration Perspective of WebSphere Integration Developer, right-click the module, and select **Open With** → **Assembly Editor**. The Assembly Diagram window appears with the modules Import component in view.

12. To create a Stand-alone Reference, click the top-most icon in the left-side (vertical) frame of the Assembly Diagram window. A new menu of icons appears.

13. Create a Stand-alone Reference. Drag the Stand-alone Reference icon into the Assembly Diagram window.

14. Create wiring from the Stand-alone Reference to FlatFileOutboundInterface. Click and drag the module's Import component to the Stand-alone reference. This draws a wire from the Import component to the new Stand-alone Reference. The Add Wire dialog box is displayed.

15. Click **OK**. The new Standalone Reference component displays in the Assembly Diagram window with a wire that connects it to the modules Import component.

16. Edit the ManagedConnectionFactory properties. In WebSphere Integration Developer, double click the first element under FlatFileOutboundModule to open the Assembly Editor.

    a. Select **Properties** → **Binding** → **Connection** → **ManagedConnectionFactory properties**

    b. Edit the ManagedConnectionFactory properties according to your specific system requirements.

    c. Click **Save** when you are finished editing the ManagedConnectionFactory properties.

17. In the Business Integration perspective, right click FlatFileOutboundModule.
18. Select **Test** → **Test Module** from the pop-up menu. The Events window is displayed.
19. In the Events window, select an operation.
    a. Populate the values in the business object based on the operation that you want to perform.
    b. To start the WebSphere Process Server and install the outbound application, click **Continue**.

The adapter application package, including all its SCA artifacts, is now configured and deployed. You can now run the sample application.

Related tasks

"Enabling logging" on page 42
The WebSphere Adapter for Flat Files maintains a log file that you can view to determine the status of event processing. All events and errors that relate to the adapter are tracked by the log file, along with the date, time, and event for each log entry. Since the adapter logs an error message when it encounters an error or warning condition, the log file is a good source to start troubleshooting problems.

"Enabling tracing" on page 44
Tracing determines what level of errors or warnings are captured in the adapter log file. You can trace messages regarding adapter processing by defining a tracing level.

Related reference

"J2C connection factory properties" on page 36
This topic describes J2C connection factories and the properties used to configure a target enterprise information system (EIS) instance. These properties affect outbound processing and correspond to the ManagedConnectionFactory interface of the J2EE Connector Architecture Specification

"Resource adapter properties" on page 40
Using the enterprise service discovery wizard when you first configure the adapter (and later, via the WebSphere Process Server administrative console), you can configure Resource adapter properties. This category of properties includes logging and tracing properties and adapter-specific properties.

Running the sample application for inbound operations

After you deploy and configure the sample application package and adapter, you can run the application to illustrate how the adapter supports outbound processing of business objects.

1. Deploy the FlatFileInboundModuleApp.ear in the WebSphere Process Server.
2. Edit the ActivationSpecification configuration using the administrative console of the WebSphere Process Server.
3. Place the required database driver in the CLASSPATH of the WebSphere Process Server.
4. Place an event file in the event directory.
5. Start the inbound application.
6. Verify that the event file was polled.

Note: The RAR file installed on your system is more recent than the one provided with the samples. If you notice any major functional discrepancies, replace the RAR file in the samples package with the installed RAR, generate the EAR with the enterprise service discovery wizard and re-deploy the EAR to the server.

Running the sample application for outbound operations

After you deploy and configure the sample application package and adapter, you can run the application to illustrate how the adapter supports outbound processing of business objects.

1. Deploy the FlatFileOutboundModuleApp.ear in WebSphere Process Server.
2. Edit the ConnectionFactory configuration using the administrative console of the WebSphere Process Server.
3. To invoke the ServiceClient, open a web browser and enter a URL to run the servlet client. For example, if the name of the dynamic web project is ″MyWebProject.″ Then, the URL would look like the following: http://localhost:9080/MyWebProject/ServiceClient
4. Enter the following and click **NewObject**.
   - **Namespace**: this should be the same as the target namespace in the .wsdl file.
   - **Operation name**: enter the operation that you would like to test. For example, Business Object.
   - **IsWrappedStyle**: set to True when you are creating a new object
5. Enter the required attributes for the selected operation and set IsWrappedType as false. Click **OK**.
6. Review the application output for confirmation of how the adapter processed the service.

   Note: The RAR file installed on your system is more recent than the one provided with the samples. If you notice any major functional discrepancies, replace the RAR file in the samples package with the installed RAR, generate the EAR with the enterprise service discovery wizard and re-deploy the EAR to the server.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION ″AS IS″ WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

IBM Corporation
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

Programming interface information, if provided, is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information 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:

i5/OS
IBM
the IBM logo
AIX
AIX 5L
CICS
CrossWorlds
DB2
DB2 Universal Database
Domino
HelpNow
IMS
Informix
iSeries
Lotus
Lotus Notes
MQIntegrator
MQSeries
MVS
Notes
OS/400
Passport Advantage
pSeries
Redbooks
SupportPac
WebSphere
z/OS

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

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

This product includes software developed by the Eclipse Project (http://www.eclipse.org/).

WebSphere Adapters, Version 6.0

Printed in USA