



Adapter for Flat Files User Guide

Note

Before using this information, be sure to read the general information in "Notices" on page 133.

28February2007

This edition applies to version 6, release 0, modification 2 of WebSphere Adapter for Flat Files (product number 5724-L78) 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.

© **Copyright International Business Machines Corporation 2006. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. About this information.	1
Chapter 2. What's new	3
New in this release	3
Release notes	3
Chapter 3. Introduction to the WebSphere Adapter for Flat Files	5
Hardware and software requirements	5
Standards compliance	5
Accessibility	5
Internet Protocol Version 6.0	6
Technical overview of the Adapter for Flat Files	6
Outbound processing	7
Inbound processing	9
Business objects	13
Enterprise service discovery	17
Globalization and bidirectional transformation.	18
Chapter 4. Planning for adapter implementation	23
Security	23
WebSphere Adapters in clustered environments	23
Roadmap for installing, configuring, and deploying the adapter.	24
Chapter 5. Installing WebSphere Adapter for Flat Files, version 6.0.2	27
Installation pre-requisites	27
Performing the installation	27
Migrating to version 6.0.2	28
Backward compatibility	28
Performing the migration.	29
Uninstalling the adapter	30
Chapter 6. Configuring the adapter for deployment	31
Creating the adapter project in WebSphere Integration Developer	31
Creating custom data bindings	33
Required folders.	34
Configuring the adapter for outbound processing	35
Generating business objects using enterprise service discovery	35
Configuring the adapter for inbound processing	43
Generating business objects using enterprise service discovery	43
Chapter 7. Deploying the module	55
Exporting the project as an enterprise archive resource (EAR) file	55
Installing the module	56
Setting or changing adapter configuration properties	57
Setting resource adapter properties	57
Setting managed (J2C) connection factory properties.	58
Setting activation specification properties for the EIS	59
Chapter 8. Configuring troubleshooting tools	61
Enabling tracing with the Common Event Infrastructure (CEI)	61
Configuring logging properties	62
Changing the log and trace file names	64
Installing or upgrading IBM Support Assistant	65

Chapter 9. Administering the adapter.	67
Starting the adapter	67
Stopping the adapter	67
Troubleshooting and support	68
Exception: XAResourceNotAvailableException	68
Self help resources	68
Contacting IBM Software Support	69
Chapter 10. Quick start tutorials.	73
Introduction	73
Tutorial 1: Outbound processing with data transformation.	74
Creating the adapter project in WebSphere Integration Developer	74
Configuring the adapter for outbound processing	75
Testing outbound with data transformation for the create operation	81
Testing outbound with data transformation for the exists operation.	82
Testing outbound with data transformation for the list operation	85
Testing outbound with data transformation for the retrieve operation	87
Testing outbound with data transformation for the append operation	89
Testing outbound with data transformation for the overwrite operation	90
Testing outbound with data transformation for the delete operation	92
Tutorial 2: Inbound processing with data transformation	93
Creating the adapter project in WebSphere Integration Developer	93
Create the event database.	95
Creating and configuring the data source	95
Configuring the adapter for inbound processing	99
Testing the assembled adapter application.	106
Tutorial 3: Splitting event files and reassembling with the append operation during inbound processing	107
Creating the adapter project in WebSphere Integration Developer	107
Configuring the adapter for inbound processing.	108
Testing the assembled adapter application.	116
Troubleshooting the tutorial	117
Chapter 11. Reference information	119
Enterprise service discovery connection properties	119
Adapter configuration properties	121
Resource adapter properties	121
Managed (J2C) connection factory properties	123
Activation specification properties	124
Adding jar files to WebSphere Integration Developer versions 6.0.1.1 and earlier	129
Settings for controlling bidirectional transformation	129
Messages	131
Related product information	131
Notices	133
Programming interface information	135
Trademarks and service marks	135
Index	137

Chapter 1. About this information

This documentation is for integration developers who implement, configure, and deploy WebSphere® Adapter for Flat Files. To use it, you should understand business integration concepts and possess certain technical skills.

Integration developers design, assemble, test, and deploy business integration solutions. This information is for those who are deploying WebSphere Adapter for Flat Files in a solution that requires data exchange between enterprise information systems (EIS) and Java™ Platform, Enterprise Edition (J2EE) applications. To use it, you should understand and have experience with the following concepts, standards, and tools:

- The business solution and environment.
- Databases, data access issues, transactional models, and connections across heterogeneous relational databases, queues, and Web services.
- Business integration mechanisms, including the Service Component Architecture (SCA) programming model and the Service Data Objects (SDO) data model.
- The J2EE standard and J2EE applications.
- The capabilities and requirements of WebSphere Process Server or WebSphere Enterprise Service Bus, depending on the host used in the environment. You should know how to configure and administer the host server and how to use the administrative console.
- The tools and capabilities provided by WebSphere Integration Developer. You should know how to use these tools to wire components and complete other integration tasks.

To complete the deployment, you should know how to perform the following tasks:

- Create required scripts, tools, and templates for both testing and deployment
- Resolve interdependencies between entities such as enterprise beans, workflows, and Web pages
- Write procedures to use database access logic efficiently
- Build data models for external data access tools
- Implement security measures

Chapter 2. What's new

WebSphere Adapter for Flat Files, version 6.0.2 provides enhancements to version 6.0 of the adapter.

New in this release

WebSphere Adapter for Flat Files, version 6.0.2 includes many enhancements such as support for multiple activation specification instances, delimiter-based file splitting, and data transformation.

New in version 6.0.2:

- Support for multiple activation specification instances that can each poll from a unique event store.
- Support for delimiter based file splitting during inbound processing.
- Support for polling from multiple event directories.
- Support for using multiple business objects.
- Support for passing event files by reference.
- Support for the Data Transformation.
- Four tutorials, which guide you through the process of creating an adapter project, generating business objects, deploying a module, and testing the module, have been added. These tutorials are self-contained, and each one can be completed in under an hour. These tutorials take the place of the samples that were documented in previous versions of the user guide.

Release notes

The release notes for WebSphere Adapter for Flat Files, version 6.0.2 summarize new features and functions in this release and document any known workarounds.

Release notes for this adapter can be found at the following Web site: [Adapter for Flat Files release notes](#).

Chapter 3. Introduction to the WebSphere Adapter for Flat Files

The IBM® WebSphere Adapter for Flat Files connects Java 2 Platform, Enterprise Edition (J2EE) components running on WebSphere Process Server or WebSphere Enterprise Service Bus with file systems running on an enterprise information system (EIS). The adapter provides a means for the J2EE component and the file system to interact. For example, the J2EE component, when configured to work with the adapter, can create a file with specified contents in the EIS file system.

Hardware and software requirements

Before installing Adapter for Flat Files, you must verify that your environment meets the necessary requirements. These requirements fall into two categories: supported platforms for running the adapter installer, and hardware and software requirements for configuring, deploying, and running the adapter.

Supported platforms for running the adapter installer

The supported platforms for running the adapter installer are located in the "Installing" section of Installing IBM WebSphere Adapters.

Hardware and software requirements for configuring, deploying, and running the adapter

The hardware and software requirements for configuring, deploying, and running the adapter are located at the following Web site: IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: software requirements. From the IBM WebSphere Adapters list, select the link for the Adapter for Flat Files, Version 6.0.2.

Standards compliance

This product is compliant with several government and industry standards, including accessibility standards and Internet protocol standards.

Accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability. The WebSphere Adapters software is fully accessible and section 508-compliant. Accessibility features enable users with physical disabilities, such as restricted mobility or limited vision, to operate software products successfully. These features are built into the installation and administration features of WebSphere Adapters.

Installation

You can install WebSphere Adapters either through a graphical user interface or silently through a script. The silent installation method is recommended for users with accessibility needs.

Administration

The administrative console of either WebSphere Process Server or WebSphere Enterprise Service Bus is the primary interface for deployment and administration of the enterprise applications. These consoles are displayed within a standard Web browser. By using an accessible Web browser, such as Microsoft® Internet Explorer or Netscape Browser, you are able to:

- Use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen
- Use voice recognition software, such as IBM ViaVoice®, to enter data and to navigate the user interface
- Operate features by using the keyboard instead of the mouse

You can configure and use product features by using standard text editors and scripted or command line interfaces instead of the graphical interfaces that are provided.

When appropriate, the documentation for specific product features contains additional information about the accessibility of the features.

Enterprise service discovery wizard

The enterprise service discovery wizard is the primary component used to create enterprise applications with the adapters. This wizard is implemented as an Eclipse plug-in that is available through WebSphere Integration Developer and is fully accessible.

Keyboard navigation

This product uses standard Microsoft Windows® navigation keys.

IBM and accessibility

See the *IBM Accessibility Center* for more information about the commitment that IBM has to accessibility.

Internet Protocol Version 6.0

IBM WebSphere Process Server relies on WebSphere Application Server for Internet Protocol Version 6.0 compatibility.

IBM WebSphere Application Server Version 6.0 and its JavaMail component support dual stack Internet Protocol Version 6.0 (IPv6).

For more information about this compatibility in WebSphere Application Server, see IPv6 support in the WebSphere Application Server information center.

For more information about IPv6, see www.ipv6.org.

Technical overview of the Adapter for Flat Files

The IBM WebSphere Adapter for Flat Files facilitates the exchange of business data in the form of delimited records in the event file between file systems and J2EE applications. The adapter supports inbound and outbound operations and the use of business objects, business components, and business services.

The adapter, which is embedded within the application server run time, facilitates communication between an enterprise information file system and various intelligent programs called endpoints. To keep track of endpoints and status, the adapter maintains an event store.

The following figure shows the adapter architecture. Arrows represent the processing flow for both inbound and outbound operations.

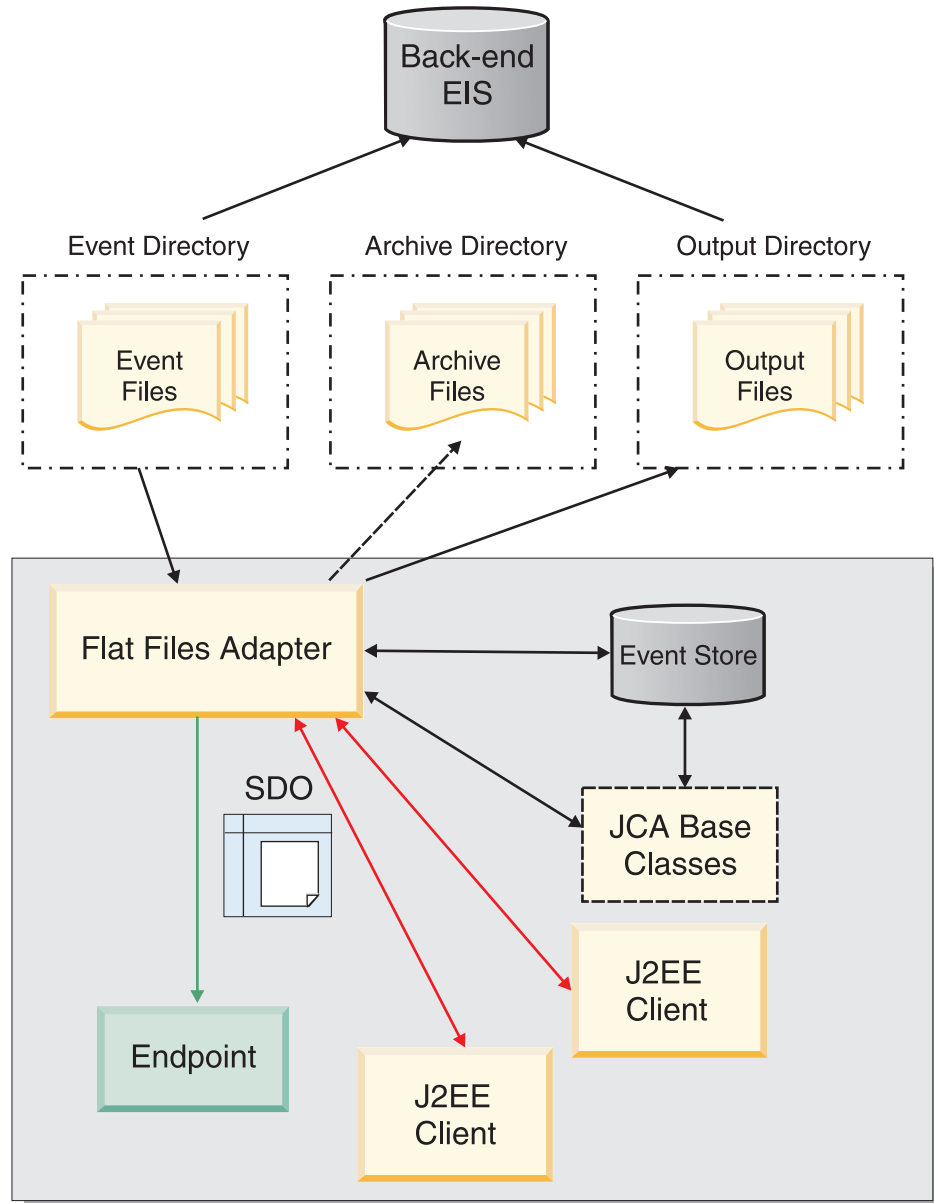


Figure 1. Architecture diagram

Outbound processing

The Adapter for Flat Files supports synchronous outbound request processing. During outbound operations, the J2EE application makes a call by sending a request, in the form of a business object, to the adapter. The adapter processes the request and based on the operation performed, the adapter may return a business object to the J2EE application.

Each request sent from a J2EE client to the adapter includes all the information required for creating the output file. This includes the directory where the output file will be created, the output file name, and the operation associated with it. The file name is treated as the key for the Flat Files request/response business object entity.

Outbound request processing consists of the following steps. The outbound service client:

1. Looks up the business object factory service.
2. Creates a business object from the business object factory.
3. Locates the adapter service.
4. Invokes the appropriate function on the adapter service by passing the function name and the business object.

Passing parameters

Passing parameters defines which operation will be performed by the adapter. Using the service client, you can pass protocol-specific parameters, such as the directory path and file name.

You can pass protocol-specific parameters in the following ways:

- Set the OutputDirectory and StagingDirectory properties in the ManagedConnectionFactory properties in the enterprise service discovery wizard before deployment or in the server administrative console after deployment.
- Set protocol-specific information (such as directory name and file name) in the custom data binding of the wrapper business object.
- Include file content as part of the input business object.

Note: The values set at the business object level override those set at the ManagedConnectionFactory level.

Supported outbound operations

WebSphere Adapter for Flat Files supports the operations described below for outbound processing.

Table 1. Supported operations

Operation	Response
Append	The content in the request is appended at the end of a file.
Create	A file with the user specified filename is created in the user specified directory and the content of the file is 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 user specified directory, a successful response is returned.
List	Return all the file names in a directory specified in the request.
Overwrite	Overwrites the file in the directory with the content specified in the request.

Table 1. Supported operations (continued)

Operation	Response
Retrieve	Returns the content of the file specified in the request.

Data Transformation Framework for outbound processing

During outbound processing, Data Transformation Framework (DTF) enables the adapter to convert the data contained in WebSphere Adapter business objects into serialized data formats such as XML. This conversion is necessary because external applications and technologies often only understand their own native or industry standard data formats. Use of DTF enables users to bridge this communication gap.

Inbound processing

The adapter supports asynchronous inbound request processing. The adapter polls the enterprise information system (EIS), pulls events from the EIS, converts the connection information and data into business objects, and sends the business objects to the configured endpoint on the application server.

During inbound processing, the adapter polls the file system at regular intervals for any events created by the back-end EIS. When events are found, the adapter sends information pertaining to the event file and any data to the application server in business object form. The following steps are performed by the adapter during inbound event processing:

1. EIS generates events in file form and stores them in an event store.
2. Adapter polls the file directory for events.
3. Adapter assigns each event an event ID.
4. Adapter reads each event file as bytes and parses the file if file spitting is enabled.
5. Adapter places the event file inside a wrapper business object and sends it to the endpoint on the application server.

Note: If file splitting is enabled, the business object contains additional information regarding the file size and the event ID.

6. After it's been confirmed that the business object reached the endpoint, the event is deleted from the event store. If archiving is enabled, the event is moved to an archiving table before it is deleted from the event store.

Event store

The adapter creates the event store or event table on the enterprise information system (EIS) during deployment to the server. Each time a file is created, updated, or deleted, the adapter tracks this as an event. The status of events is continually updated by the adapter for recovery purposes until they are delivered to a configured endpoint on the application server.

During inbound processing, the adapter polls event files from the event directory at regular intervals. Before posting each event to the endpoint, the adapter creates an entry for the event in the event store and keeps track of it by updating the status of the entry as it moves through the system. If an event is successfully posted, event store entries are deleted. For failed events, the entries remain in the event table. Optionally, the adapter can archive successfully polled event files in a

user specified archive directory. The event directory, archive directory, poll interval, and poll quantity (the number of event files to poll in a single poll cycle) are all configurable parameters.

The status of each event is stored for recovery purposes. The figure below illustrates the event management framework.

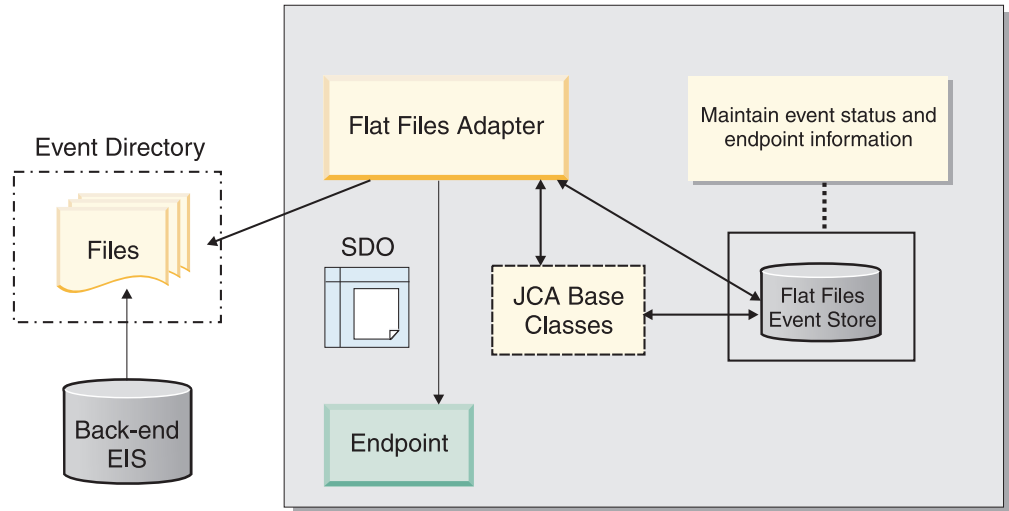


Figure 2. Event management framework

Event store structure:

The event store is used by the adapter to track events. The following table notes which values are stored for each event.

Table 2. Event table structure

Column name	Type (length)	Description
EVNTID	Varchar (255)	Used to Track events during inbound processing. Each event requires an event ID for tracking purposes. This must be a unique identifier in the table.

Table 2. Event table structure (continued)

Column name	Type (length)	Description
EVNTSTAT	Integer	<p>The status of the event. The adapter uses the status to determine whether an event is new or in process.</p> <p>Event status values:</p> <p>NEW(0)</p> <p>The event is ready to be processed.</p> <p>PROCESSED (1)</p> <p>The adapter successfully processed and delivered the event.</p> <p>FAILED (-1)</p> <p>The adapter was unable to process this event due to one or more problems.</p>
XID	Varchar(255)	Used by the adapter for assured event delivery and recovery.
EVNTDATA	Varchar(255)	Used to track failed events so that they will not be processed again during recoveries. Failed events are marked "ARCHIVED."

Event archival values:

The adapter can be configured to archive processed event files in a user-configured directory. The success or failure of an archived event is noted in the file extension.

All archived events in the user configured Archive Directory are stored with a "PROCESSED" file extension. The extensions of the files for SUCCESS and FAILURE are configurable based on the following activation specification properties: FailedArchiveExt, OriginalArchiveExt, and SuccessArchiveExt.

The following table lists the archive extensions used by the adapter.

Table 3. Event archive values

Extension	Definition	Format
SUCCESS	The event file was delivered to the endpoint.	<filename>_<timestamp>.SUCCESS
FAIL	The event file was not delivered to the endpoint.	<filename>_<timestamp>.FAIL

File splitting

If the adapter is routinely retrieving large files from the enterprise information file system, you can choose to chunk the files into smaller chunks using the file splitting functionality of the adapter. When file splitting is enabled, event files will

be split into several chunks and posted to the endpoint separately. The adapter does not reassemble the chunks at the end point, however the adapter provides the required information in the business graph to reassemble the chunks.

Files are split by the adapter based on the value specified in the SplitCriteria property. This value can be either a delimiter or a file size. Since file splitting is an optional feature, it can be disabled by leaving the value of the SplitCriteria property and the SplittingFunctionClassName property empty when configuring the activation specification properties with the enterprise service discovery wizard or by setting the SplitCriteria property to zero.

Note: If the value for EventContentType is null, then the SplitCriteria property is automatically configured to split files based on size.

Delimiter file splitting

When one or more characters such as commas (,), semicolon (;), quotes (" , '), braces ({}) or slashes (/ \) are used to separate a business object, the adapter can use this information to split files into smaller chunks. These chunks are forwarded to the server separately and then reassembled by the Event Sequencing feature of WebSphere Process Server. This is called delimiter file splitting and it can be enabled using the SplitCriteria activation specification property.

When event files are split into such chunks, it is important to remember that each chunk creates a business object. This means that the value specified for the PollQuantity property and the number of business objects obtained at the endpoint can be different. When file splitting based on a delimiter is enabled, the PollQuantity property specifies the number of such event files that are present in the Event Store and the class used to split the event file is set in the Splitting Function Class Name activation specification property.

To demonstrate how the PollQuantity value works with delimiter file splitting, consider two event files. The first event file contains a business object and the second file contains two business objects. If the PollQuantity value is 2, then the first business object from the first event file and the next business record from the second event file are sent in the first poll cycle. The second business object from the second file is sent in the next poll cycle.

The following rules apply to the use of delimiters:

- All new lines in the delimiter are represented as \n. The adapter translates the \n to the platform-specific new line character.
- If there is more than one delimiter, each delimiter must be separated by a semicolon (;). If the semicolon (;) is part of the delimiter, the semicolon (;) must be escaped as \;. For example, if the delimiter is ##\;## then it is processed as ##;\##, which means that the semicolon (;) is part of the delimiter.
- To skip content that is part of the delimiter, specify a double semicolon (;;) so that the content is skipped between the delimiters. For example, if the event file contains a business object in the below format and the delimiter is ##;\;\$\$, then:

```
Name=Smith
```

```
Company=IBM
```

```
##this is the content which will be skipped by the adapter$$
```

The adapter will consider the delimiter as ##;\;\$\$ and skip "this is the content which will be skipped by the adapter".

- The delimiter takes any value and there are no restrictions. It is a combination of `\n` and a semicolon (`;`) if there is more than one delimiter. A delimiter does not have to be a composition of `\n` and `;` at all times. `\n` is used only when a newline is to be considered while splitting the contents of the file. The following delimiters are valid:
 - `####;\n;\n`
 - `####;$$$$;\n;####`
 - `%%%;$$$$;#####`
 - `\n;\n;$$$$`
 - `####\;#####;\n;$$$$`
 - `\n;\n;\n`
 - `####;$$$$`
- If the delimiter is located at the end of the file, `SplitCriteria` takes `END_OF_FILE` which means that the business object is the physical end of the file.

File splitting by size

File splitting by size is performed based on the value specified in the `SplitCriteria` property. If the event file size is greater than the value specified in the `SplitCriteria` property, the files is split into chunks and each chunk is posted to the endpoint separately. If the event file size is less than the `SplitCriteria` value, then the entire event file is posted to the endpoint. When event files are split into chunks, it is important to remember that each chunk creates a business object. This means that the value specified for the `PollQuantity` property and the number of business objects obtained at the endpoint can be different. While the adapter polls based on the `PollQuantity` value, the adapter actually processes the number of business objects in the file one at a time. For example, if an event file is chunked into three parts, then one file will be polled and three business objects will be received by the endpoint (since each chunk creates a single business object).

At the endpoint, the adapter does not reassemble chunked data into a single file. Rather, this functionality is handled by the Event Sequencing feature of WebSphere Process Server. However, the Flat Files adapter does provide information about a chunk that enables WebSphere Process Server to reassemble the chunks into a single file. The chunk information is included in the `chunkFileName` property of the `FlatFile` wrapper business object. The chunk information includes the chunk size in bytes and the event ID. The event ID of a chunk uses the following form: `eventFileLocation/_timestampStr/_MofN`, where `M` is the current chunk number and `N` is the total number of chunks. An example event ID would look like the following: `C:\flatfile\eventdir\eventfile.in/_2005_01_10_10_17_49_864/_3of5`, where `timestampStr` has the following format: `year_month_day_hour_minutes_seconds_milliseconds`.

Data Transformation Framework for inbound processing

During inbound processing, Data Transformation Framework (DTF) enables the adapter to convert the event data to a WebSphere Adapter business object. This conversion is necessary because the service components in WebSphere Process Server can only consume WebSphere Adapter business objects.

Business objects

Business objects carry the functional properties, data transformation information, and file content that the adapter needs to process requests and generate responses. Depending on your business need, you can use the default business objects created

by the enterprise service discovery wizard or you can import custom business object definitions (xsd files) that you have stored on your local system.

Business object naming conventions

Business object names should reflect the structure they represent, such as Customer or Address. Names will most likely be derived during the metadata import process of enterprise metadata discovery, based on the name given by the enterprise information system (EIS).

Business object names should be converted to camel case, in which separators such as spaces and underscores are removed, and the first letter of each word is capitalized; for example, ORDER_LINE_ITEM would be converted to OrderLineItem.

The parent business object graph should be named for the contained business object, followed by BG; for example, CustomerBG for a Customer business object.

Business object names have no semantic value to the adapter or the database.

Business object structure

The adapter's business object structure is based on the generic WebSphere Business Integration business object structure, which is modeled as a base XML schema. The adapter defines and generates the business objects during enterprise service discovery.

Generic FlatFileBG object

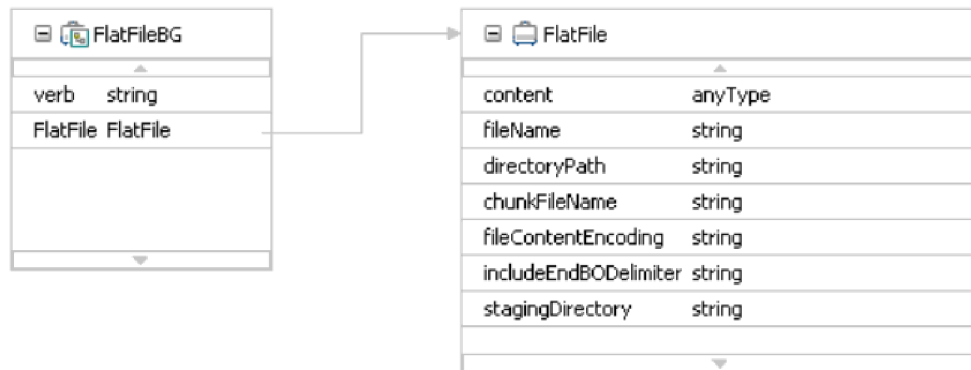


Figure 3. The generic FlatFileBG business object structure

CustomerWrapperBG object

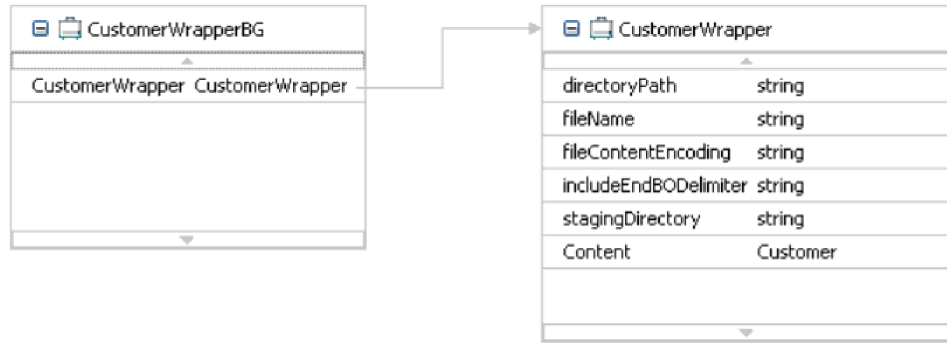


Figure 4. The CustomerWrapperBG business object structure

Retrieve operation business object

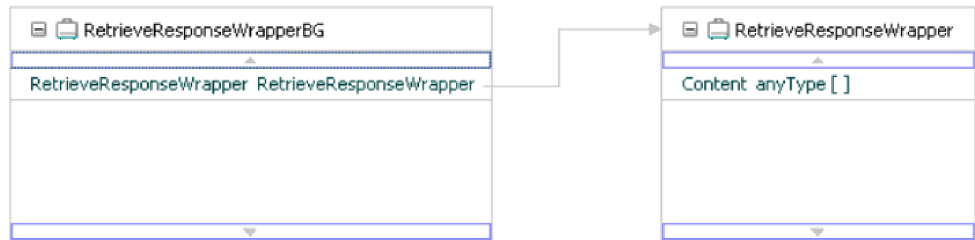


Figure 5. Structure of the Retrieve operation business object

List operation business object

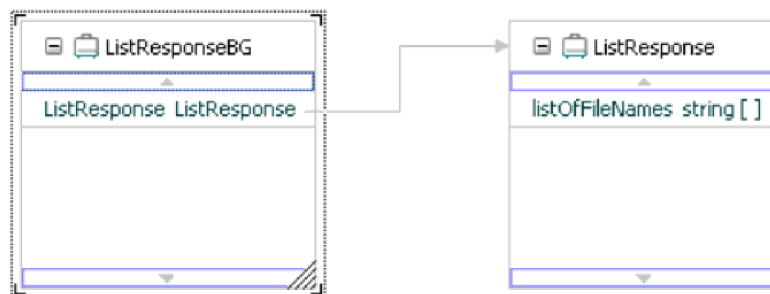


Figure 6. Structure of the List operation business object

Exists operation business object

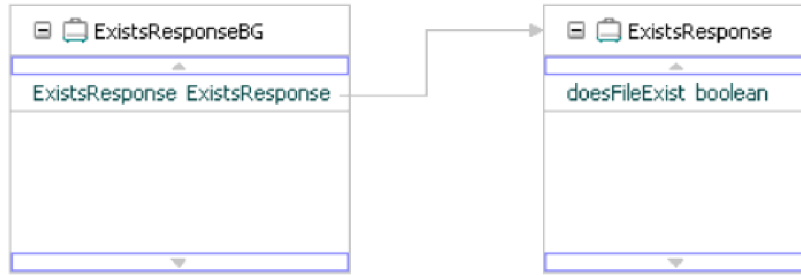


Figure 7. Structure of the Exists operation business object

Attribute properties

Business object architecture defines various properties that apply to attributes. This section describes how the adapter interprets these properties.

The following table, "Attribute properties", describes these properties.

Table 4. Attribute properties

Attribute property	Description
Cardinality	Each business object attribute that represents a child or an array of child business objects has the value of single (1) or multiple (n) cardinality. Only single cardinality flat business objects are supported.
Key and foreign key	These attributes are not used by the adapter.
Name	Represents the unique name of the attribute.
Required	This attribute is not used by the adapter.
Special	None.
Type	The attribute type can be either simple or complex. Simple types are: Boolean, String, LongText, Integer, Float, Double and Byte[]. A typical complex type is another business object type.

Supported operations

The adapter uses operations to specify the action to be performed on the enterprise information file system during outbound processing. Refer to the supported operations table below for a list of the supported operations and the expected responses for each.

Table 5. Supported operations for outbound processing

Operation	Response
Append	The content in the request is appended at the end of a file.
Create	A file with the user specified filename is created in the user specified directory and the content of the file is sent across in the request.
Delete	Deletes the file from the directory specified in the request.

Table 5. Supported operations for outbound processing (continued)

Operation	Response
Exists	If the file in the request exists in the user specified directory, a successful response is returned.
List	Returns all the file names in a directory specified in the request.
Overwrite	Overwrites the file in the directory with the content specified in the request.
Retrieve	Returns the content of the file specified in the request.

Enterprise service discovery

The enterprise service discovery wizard is a tool you use to configure your adapter before it is deployed to WebSphere Process Server or WebSphere Enterprise Service Bus. Enterprise service discovery connects to the enterprise information file system, discovers services (based on search criteria you provide), and generates business objects and interfaces.

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 of business graphs and business objects.

The following figure illustrates the enterprise service discovery wizard flow. When finished, an EAR file containing all of the information for your adapter project is created. This EAR file can then be deployed to the application server.

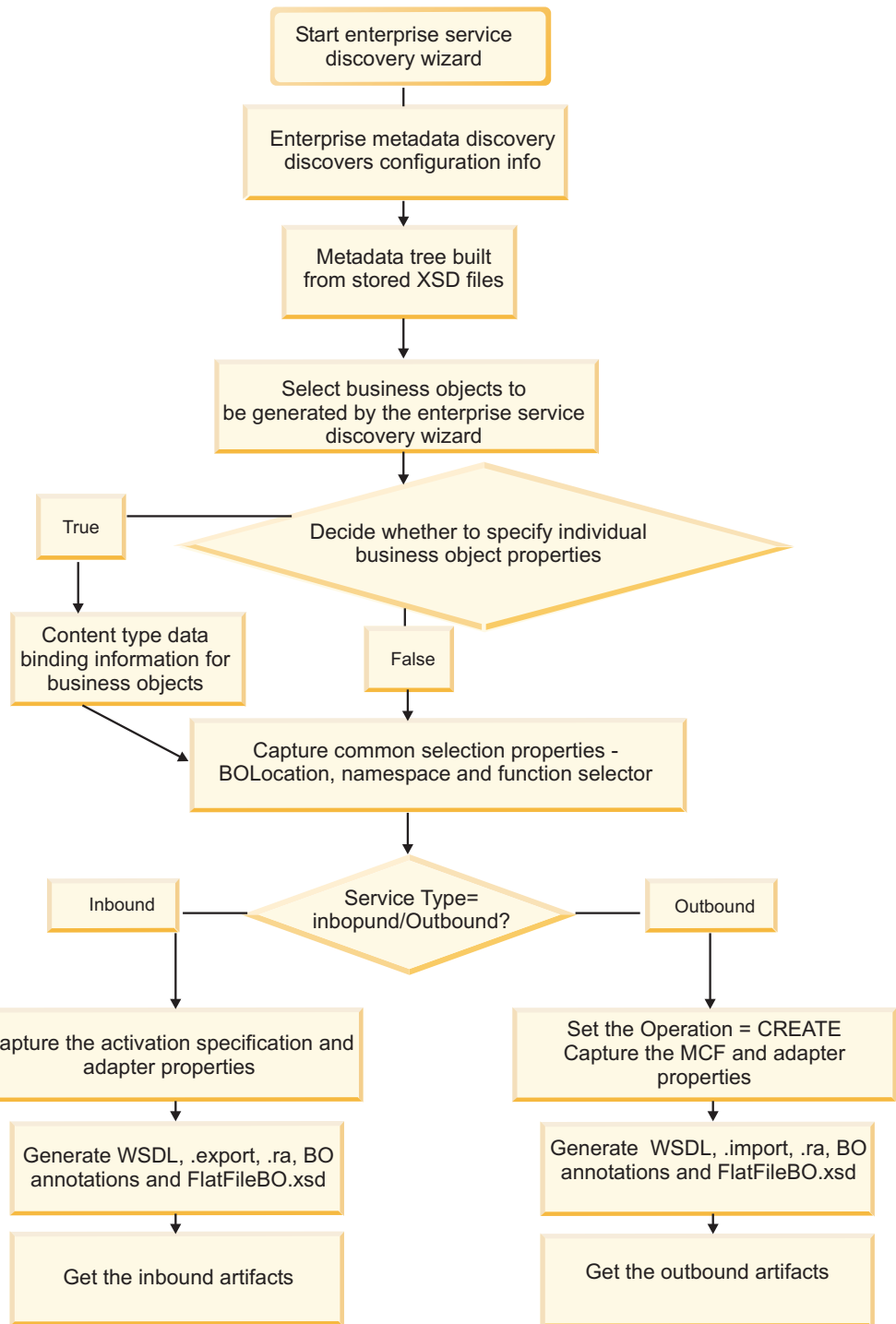


Figure 8. Basic enterprise service discovery wizard flow

Globalization and bidirectional transformation

The adapter is globalized to support single- and multi-byte character sets and deliver message text in the specified language. The adapter also performs bidirectional transformation, which refers to the task of processing data that

contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Globalization

The Java 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). Components in the WebSphere Business Integration system are written in Java. Therefore, when data is transferred between 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 region, the adapter uses the locale of the system on which it is running.

Bidirectional transformation

Languages such as Arabic and Hebrew are written from right to left, yet they contain embedded segments of text that are written left to right, resulting in bidirectional script. When software applications handle bidirectional script, standards are used to display and process it. WebSphere Process Server and WebSphere Enterprise Service Bus use the Windows standard format, but an enterprise information system exchanging data with WebSphere Process Server or WebSphere Enterprise Service Bus can use a different format. WebSphere Adapters transform bidirectional script data passed between the two systems so that it is accurately processed and displayed on both sides of a transaction.

Bidirectional format

WebSphere Process Server and WebSphere Enterprise Service Bus use the bidirectional format of ILYNN (implicit, left-to-right, on, off, nominal). This is the format used by Windows. If an enterprise information system uses a different format, the adapter converts the format prior to introducing the data to WebSphere Process Server or WebSphere Enterprise Service Bus.

Five attributes comprise bidirectional format. When you set bidirectional properties, you assign values for each of these attributes. The attributes and settings are listed in the following table.

Table 6. Bidirectional format 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

Table 6. Bidirectional format attributes (continued)

Letter position	Purpose	Values	Description	Default setting
5	Numeric Shaping	H C N	Hindi Contextual Nominal	N

The adapter transforms data into a logical, left-to-right format before sending the data to WebSphere Process Server or WebSphere Enterprise Service Bus.

Using bidirectional properties

You can use multiple bidirectional properties to control the transformation of both content data and metadata. You can set special bidirectional properties to exclude either content data or metadata from bidirectional transformation, or to identify data that requires special treatment during a transformation.

The following table describes four types of bidirectional properties.

Table 7. Bidirectional property types

Property type	Data transformations
EIS	Controls the format for content data, or data that is sent by the enterprise information system.
Metadata	Controls the format for metadata, or data that provides information about the content data.
Skip	Identifies content or metadata to exclude from transformation.
Special Format	Identifies certain text, such as file paths or URLs, that requires different treatment during the transformation process. Can be set for either content data or metadata.

You can set properties that control bidirectional transformation in three areas.

- **Resource adapter properties:** These properties store default configuration settings, including the TurnBiDiOff property, which controls whether the adapter instance performs bidirectional transformation or not. Use the administrative console of the server to configure these properties.
- **Managed (J2C) connection factory properties:** These properties are used at run time to create an outbound connection instance with an enterprise information system. Once the managed connection factory properties are created, they are stored in the deployment descriptor.
- **Activation specification properties:** These properties hold the inbound event processing configuration information for a message endpoint. Set them as you perform enterprise service discovery, or use the administrative console of the server.

Business object annotations

Some adapters allow you to annotate bidirectional properties within a business object. Do this to add information that specifically controls the transformation of a business object or part of a business object. Use business object editor, a tool within WebSphere Integration Developer, to add annotations at these levels:

- Business object
- Business object application-specific attribute
- Business object attribute
- Business object attribute application-specific attribute

Property scope and lookup mechanism

After you set values for bidirectional properties for an adapter and annotate business objects where appropriate, the adapter performs bidirectional transformations. It does so by using logic that relies on a hierarchical inheritance of property settings and a lookup mechanism.

Properties defined within the resource adapter are at the top of the hierarchy, while those defined within other areas or annotated within a business object are at lower levels of the hierarchy. So for example, if you only set values for EIS-type bidirectional properties for the resource adapter, those values are inherited and used by transformations that require a defined EIS-type bidirectional property whether they arise from an inbound (activation specification) transaction or an outbound (managed connection factory) transaction.

However, if you set values for EIS-type bidirectional properties for both the resource adapter and the activation specification, a transformation arising from an inbound transaction uses the values set for the activation specification.

The processing logic uses a lookup mechanism to search for bidirectional property values to use during a transformation. The lookup mechanism begins its search at the level where the transformation arises and searches upward through the hierarchy for defined values of the appropriate property type. It uses the first valid value it finds. It searches the hierarchy from child to parent only; siblings are not considered in the search.

Chapter 4. Planning for adapter implementation

Before you begin your installation, you must consider several factors, such as your adapter environment, security and performance needs, and whether you need local and globalization support.

Security

The WebSphere Adapter for Flat Files is Java 2 security enabled. In addition, you can configure additional security permissions by altering the application server's WAS.policy file and storing it in the meta-inf folder. For more details on configuring security details, see the security documentation for WebSphere Process Server.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying the WebSphere adapter enterprise archive (EAR) module to a clustered server environment. The adapter instance within the EAR module is replicated across federated servers.

WebSphere Process Server and WebSphere Application Server Network Deployment support clustered environments. Clusters are groups of servers that are managed together to balance workloads and to provide high availability and scalability. When you set up a server cluster, you create a Deployment Manager profile. The HAManager, a subcomponent of the Deployment Manager, notifies the JCA container to activate the adapter instance. The JCA container provides a runtime environment for adapter instances. For more information about clustered environments, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_v61.html.

In clustered environments, adapter instances can handle both inbound and outbound operations.

High availability for inbound operations

Inbound operations are based on events triggered as a result of updates to data in the enterprise information system (EIS) application. The adapter is configured to detect updates through event listeners or by polling an event table. The adapter then publishes the event to its endpoint.

In a clustered environment, two or more adapter instances might detect the same event. This scenario raises the possibility of duplicate event processing or data infidelity. For example, if two adapter instances are simultaneously polling the same event table with the same event type filter, one may alter data that the other adapter instance depends on, or it might fail. There is a parallel risk for event-listening adapter architectures in a clustered environment.

To avoid this condition, the HAManager for the inbound adapter instances enforces a singleton behavior. Even though all the adapter instances are started, only one of the instances detects and publishes an event to the endpoint for each type of EIS application.

When you deploy an adapter module to a cluster, the JCA container checks the `enableHASupport` property of the `ResourceAdapter` bean. If the value for the `enableHASupport` property is true, the JCA container registers all of the adapter instances with `HAManager` with a policy 1 of N. This policy means that only one of the clustered servers starts event polling (or listening) for this adapter instance. Although other adapter instances in the cluster are started, they remain dormant with respect to the active event until the active adapter instance finishes processing the event. If the server on which the polling thread was started shuts down for some reason, an adapter instance that is running on one of the backup servers is activated.

High availability for outbound operations

In clustered environments, multiple adapter instances are available to perform outbound requests. Accordingly, if your environment has multiple applications that interact with the same WebSphere adapter for outbound requests, then you might improve performance by deploying the adapter module to a clustered environment.

WebSphere Application Server Network Deployment has a workload management capability that distributes the outbound processing among the adapter instances. As a result, outbound operations in a clustered environment are similar to those in a single server environment: one adapter instance processes only one outbound request at a time. For more information on workload management, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Note: Adapter instances are replicated in a clustered server environment. When the `enableHASupport` property is set to true, which is the default setting, only one of the replicated adapter instances actively polls for events while other instances are in standby mode. If the `enableHASupport` property is set to false, all of the adapter instances replicated on cluster members actively poll for events. This may result in event duplication. Do not change the value of `enableHASupport` to false for single server environments. For information on changing the value of this property, see the `Resource adapter properties` section in this documentation. To determine whether adapter replication is supported in a clustered environment, see the `software and hardware requirements` section of this documentation.

Roadmap for installing, configuring, and deploying the adapter

Before you can use the adapter in a runtime environment, you must install, configure, and deploy it. Understanding these tasks at a high level helps you perform the steps that are needed to accomplish each task.

After successfully installing the WebSphere Adapter, you configure it using WebSphere Integration Developer. You then deploy it as an enterprise archive (EAR) file to WebSphere Process Server or WebSphere Enterprise Service Bus. The following figure illustrates this flow of tasks, and the steps that follow the figure describe each task at a high-level. For detailed instructions on installing, see *Installing IBM WebSphere Adapters*. For information about configuring and deploying the adapter, see the adapter documentation.

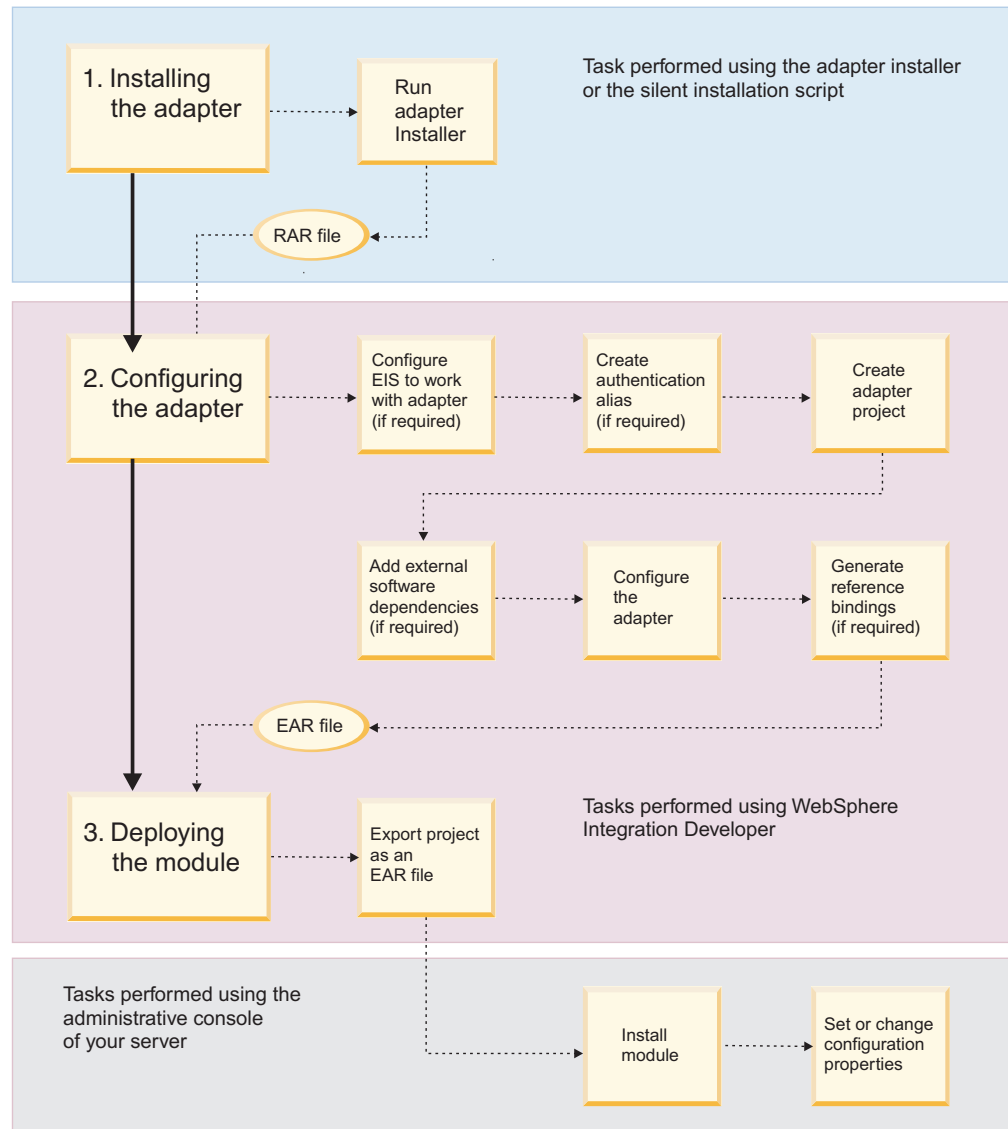


Figure 9. Roadmap for installing, configuring, and deploying the adapter

1. Installing the adapter

- a. Use the installer (a graphical user interface) or a script that runs a silent installation. Either method installs a resource adapter archive (RAR) file on your workstation. You use this RAR file to configure the adapter.

2. Configuring the adapter

- a. (If required) Configure the enterprise information system (EIS) to work with your adapter. You perform this step from within the EIS application.
- b. (If required) Create an authentication alias to access the application.
- c. Create an adapter project in WebSphere Integration Developer (J2EE Perspective) by importing the adapter RAR file.
- d. (If required) Using WebSphere Integration Developer, add any external dependencies required by your adapter to the adapter project. These dependencies are also required as part of the bundled EAR file, which is exported when you deploy the adapter.

- e. To configure the adapter, run the enterprise service discovery wizard from the Business Integration Perspective of WebSphere Integration Developer. The enterprise service discovery wizard generates business integration components and allows you to enter all the information necessary to configure the adapter for the first time. The output from the enterprise service discovery tool is saved to a business integration module project, which contains the business object, or objects, and the import or export file.
- f. (If required) Use WebSphere Integration Developer to generate reference bindings for the component created by the enterprise service discovery wizard.

3. Deploying the module

- a. From the J2EE perspective in WebSphere Integration Developer, export a business integration module project as an EAR file.
- b. Install the module on WebSphere Process Server or WebSphere Enterprise Service Bus.
- c. (If required) In the server administrative console, set (or change) the following properties:
 - Resource adapter properties
 - Managed (J2C) connection factory properties
 - Activation specification properties for the EIS

Chapter 5. Installing WebSphere Adapter for Flat Files, version 6.0.2

To install the adapter, you must check system prerequisites, perform migration steps, then perform the installation steps common to all adapters.

Installation pre-requisites

Before installing Adapter for Flat Files, you must verify that your environment meets all of the necessary hardware and software requirements. These requirements fall into two categories: supported platforms for running the adapter installer, and hardware and software requirements for configuring, deploying, and running the adapter.

Supported platforms for running the adapter installer

The supported platforms for running the adapter installer are located in the "Installing" section of Installing IBM WebSphere Adapters.

Hardware and software requirements for configuring, deploying, and running the adapter

The hardware and software requirements for configuring, deploying, and running the adapter are located at the following Web site: IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: software requirements. From the IBM WebSphere Adapters list, select the link for the Adapter for Flat Files, Version 6.0.2.

Performing the installation

The steps for installing the adapter are the same for all WebSphere Adapters. You can install the adapter either by using a graphical user interface or by performing a silent installation.

Before you begin

Review the installation prerequisites.

About this task

To install the adapter, use the following procedure.

How to perform this task

1. Install the adapter using the basic installation instructions, which are common to all adapters. These steps are located in the "Installing" section of Installing IBM WebSphere Adapters in the IBM WebSphere Business Integration information center.

Note: Some WebSphere Adapters require you to perform additional steps specific to your adapter to complete the installation. The WebSphere Adapter for Flat Files does not have this requirement.

2. After performing the basic installation steps, you can configure the adapter.

Result

The resource adapter archive (RAR) file is copied to the workstation where the adapter is installed. If you accepted the default installation location, the RAR file is placed in the following directory: C:\Program Files\IBM\ResourceAdapters\
<adapter_name>\adapter*<adapter_name>*\deploy\
CWY<adapter_variable>_<adapter_name>.rar.

What to do next

Use WebSphere Integration Developer to configure the adapter.

Migrating to version 6.0.2

To migrate from an earlier version of WebSphere Adapter for Flat Files, become familiar with the features that are being deprecated in this release.

Backward compatibility

For the 6.0.2 version of the WebSphere Adapter for Flat Files, significant changes have been made to the business object structure in order to add greater functionality to the adapter. Backward compatibility is supported in the 6.0.2 version of the Adapter for Flat Files. The older business object structures and service descriptions supported in the previous version of the adapter are backward compatible with the 6.0.2 version of the Adapter for Flat Files.

In order to provide backward compatibility, the Adapter for Flat Files supports two activation specifications. The FlatFileActivationSpec supports the 6.0.0 version of the adapter while the FlatFileActivationSpecWithXid supports the 6.0.2 version of the adapter.

Some of the properties that are exclusive to the 6.0.0 version of the adapter are not applicable to the 6.0.2 version of the adapter. There is no scenario to run the enterprise service discovery wizard for the older 6.0.0 properties and whenever the enterprise service discovery wizard is run for the inbound application, it is with respect to only 6.0.2 attributes. 6.0.2 features are not available unless the user decides to run the enterprise service discovery wizard to generate new activation specifications, managed connection factories, and interaction specifications.

Note: To upgrade from a version 6.0.0 RAR to a 6.0.2 RAR adapter file, you must use WebSphere Integration Developer, Version 6.0.1.2. Otherwise you may be unable to view the Binding properties in the Properties Tab of WebSphere Integration Developer in the Business Integration perspective.

The inbound properties that are exclusive to 6.0.0 version of the adapter are the following:

- FFEventTableName
- FFDatabaseName
- FileSplitThreshold
- FileChunkSize
- AutocreateEDT
- EDTDatabaseName
- EDTDriverName
- EDTTableName

- EDTUserName
- EDTUserPassword

For outbound processing, the OutputFileName property has been introduced in the 6.0.2 version of the adapter in order to provide backward compatibility for the 6.0.0 version of the adapter. This property will be available when you use the enterprise service discovery wizard for outbound processing.

Note: All pending inbound and outbound operations should be completed by the 6.0.0 version of the Adapter for Flat Files before performing the upgrade procedure. In addition, all event producing applications must be stopped before performing the upgrade procedure.

Performing the migration

To update an existing adapter application to include the newly installed adapter, you replace the RAR file from a previous version of the adapter with the RAR file installed as part of the WebSphere Adapter for Flat Files version 6.0.2 installation.

Before you begin

Make sure you have installed the new version of Adapter for Flat Files and have determined that your existing adapter application must be updated.

About this task

The way that you upgrade an adapter application depends on whether the application has already been deployed to WebSphere Process Server or WebSphere Enterprise Service Bus or whether the application is in the test environment of WebSphere Integration Developer. If the application has already been deployed, you use the administrative console to replace the adapter RAR file in the application. If the application is in the test environment of WebSphere Integration Developer, you import the new RAR file into the adapter project.

How to perform this task

- If your adapter application has already been deployed to WebSphere Process Server or WebSphere Enterprise Service Bus, use the following procedure.
 1. From the administrative console of the server, click **Applications** → **Enterprise Applications**, and select the enterprise application that uses the adapter to be upgraded.
 2. From the Configuration page, select **Connector Modules**.
 3. On the Connector Modules page, select the check box next to the name of the RAR file, and then click **Update**.
 4. Select **Single Module, Local File System**, and browse for the new adapter RAR file. Then click **Next**.
 5. Click **Next** on the following page.
 6. Click **Continue**.
 7. On the Install New Application page, select **Step 4: Summary**, and then click **Finish**.
 8. Click **Save changes to Master Configuration**.
 9. Click **Save**.
- If your adapter application is in a WebSphere Integration Developer test environment, use the following procedure.

1. Select the connector project to be updated
2. Import the new adapter RAR file.
3. Build and deploy dependent applications as necessary.

Result

The adapter application contains the updated (WebSphere Adapter for Flat Files version 6.0.2) RAR file.

Uninstalling the adapter

The steps for uninstalling the adapter are the same for all WebSphere Adapters. You can uninstall the adapter either by using a graphical user interface or by performing a silent uninstallation.

About this task

Uninstalling the adapter may be a required task for troubleshooting an installation problem. The steps for uninstalling the adapter are located in the "Uninstalling" section of Installing WebSphere Adapters.

Note: If you need to uninstall an adapter that is already deployed, refer to the "Additional adapter-related information you might need" section of "Related product information" on page 131.

Chapter 6. Configuring the adapter for deployment

To configure WebSphere Adapter for Flat Files so that it can be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, use WebSphere Integration Developer to create an adapter project, add required files to the project, and specify the business objects you want to discover and the system on which you want to discover them.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating and deploying a module, you create an adapter project. The adapter project contains the adapter itself plus other related artifacts. You create the project by importing the RAR file, which was copied to your local file system during installation, into WebSphere Integration Developer.

About this task

Create an adapter project (called a *connector project* in WebSphere Integration Developer) to contain the adapter (which you import from the adapter installation directory) as well as artifacts related to it. All projects are self-contained; they do not refer to objects outside of the project.

To create an adapter project, use the following procedure.

How to perform this task

1. If WebSphere Integration Developer is not currently running, start it now.
 - a. Click **Start** → **Programs** → **IBM WebSphere** → **Integration Developer 6.0**.
 - b. If you are prompted to specify a workspace, accept the default value.
The workspace is a directory where WebSphere Integration Developer stores your project.
 - c. When the WebSphere Integration Developer window is displayed, close the Welcome page.
2. Switch to the J2EE perspective:
 - a. Click **Window** → **Open Perspective** → **Other**.
 - b. Click **J2EE**.
If **J2EE** is not displayed in the Select Perspective window, select the **Show all** check box, click **J2EE**, and click **OK**.
 - c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.
 - d. Click **OK**.
3. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

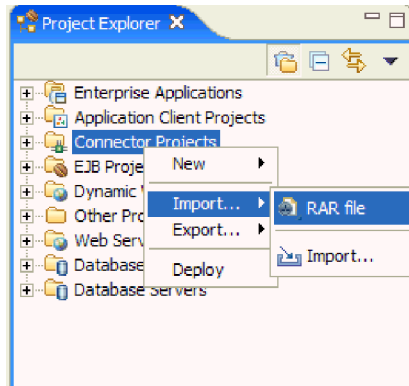


Figure 10. Importing the RAR file

4. From the Connector Import window, click **Browse** and navigate to the directory in which Adapter for Flat Files was installed.
5. Click **CWYFF_FlatFile.rar**.
The connector project has the same name as the RAR file.
6. Optional: In the **Connector project** field, type another name for the project or accept the default value.
7. Optional: In the **Target server** field, select the server to which the adapter will be deployed or accept the default value.
8. Clear the **Add module to an EAR project** check box.

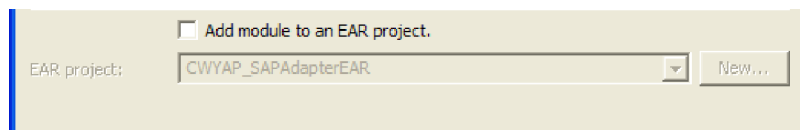


Figure 11. Clearing the Add module to an EAR project check box

Notice that the EAR project field becomes unavailable after you remove the check mark.

9. Click **Finish**.

Result

A new J2EE Connector project is created. To see its contents, expand the project in Project Explorer. For example, if the connector project is named CWYFF_FlatFile, expand **CWYFF_FlatFile**.

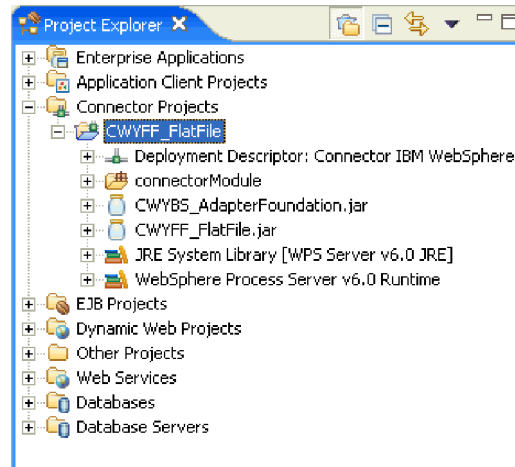


Figure 12. Flat Files connector project

Creating custom data bindings

The adapter provides an option for creating custom data bindings to meet your environment needs. A custom data binding is a Java class that you write to convert a stream of data to a business object during inbound processing and a business object to a stream of data during outbound processing. The enterprise information system (EIS) and JMS exports and imports can be configured to invoke custom data bindings.

About this task

The following steps guide you through the process of creating custom data bindings.

How to perform this task

1. Implement the custom data binding using the `com.ibm.connector.runtime.RecordHolderDataBinding` interface. This interface is located in the `com.ibm.connector.jar` file, which is contained in the WebSphere Process Server runtime libraries.
2. Create the custom data bindings using either the inbound or outbound information shown below.
 - To convert a stream of data to a business object for inbound processing, the adapter calls the public void `setRecord(Record arg0)` and public `DataObject getDataObject()` methods of the custom data binding. It first calls `setRecord` and then calls `getDataObject`.
 - a. The `setRecord` method takes an `com.ibm.j2ca.base.UnstructuredRecord` as a parameter. This class is part of the Adapter Foundation class jar file. The business object record, as text or `byte[]`, is sent to the custom data binding in this method. Other required information is also sent.
 - b. The `getDataObject()` gets the business object record, set as mentioned above, does the necessary data transformation and returns the required business object.
 - c. The `getRecordName()` present in the unstructured record gets the fully qualified namespace with which we can instantiate the business object

and extract (as mentioned in step 3) any required ASI present in the annotation. This ASI is used by the data binding. For example:

```
recordName = http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/customerwrapperbg/CustomerWrapperBG
```

```
namespace = http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/customerwrapperbg
```

```
object name = CustomerWrapperBG
```

- To convert a business object to a stream of data for outbound processing, the adapter calls the public void setDataObject(DataObject dataObject) and public Record getRecord() methods of the custom data binding. It first calls setDataObject and then calls getRecord.
 - a. The setDataObject method takes the business object as a parameter. The output of getRecord must be of type com.ibm.j2ca.base.UnstructuredRecord.
 - b. Application-specific information is obtained from the input business object as mentioned in step 3.
 - c. The getRecord method does the necessary transformation and returns the native data in the form of an unstructured record.

An example of EIS native data with delimited text is:

```
Employee~Create~sarath~IBM~Bangalore~560071
```

Business object schema:

```
complexType name="Employee"
  sequence maxOccurs="1" minOccurs="1"
    element maxOccurs="1" minOccurs="0" name="name" type="string"/
    element maxOccurs="1" minOccurs="0" name="company" type="string"/
    element maxOccurs="1" minOccurs="0" name="city" type="string"/
    element maxOccurs="1" minOccurs="0" name="zip" type="string"/
  /sequence
/complexType
```

3. To access application-specific information in the annotation of the business object, use the CWYBS_AdapterFoundation.jar file, which contains a class named com.ibm.j2ca.base.AdapterBOUtil. This class contains the method to extract the application-specific information from the business object.
4. Use the business object application programming interface (API) methods to extract each value contained in the annotation. The output is a business object representing the application-specific information.
 - For inbound, instantiate the business object from the recordName value contained in the unstructured record.
 - For outbound, call the getMetadataForObject() by passing the business object as a parameter.

What to do next

Configure the custom data bindings using the detailed instructions located in `WPS_install_directory\samples\doc\CustomEISDataBinding\index.html`.

Required folders

Before you begin to create your inbound or outbound adapter projects, you must create folders on the enterprise information system (EIS) or a local drive for events, staging, and output.

The event directory is used to store events for inbound operations. The adapter will poll this folder at regular intervals and send any found events, in the form of

business objects, to the server. The value for this directory is noted by you on the Generate artifacts screen of the enterprise service discovery wizard.

The output directory is used by the adapter to write the final output files for create, append, and overwrite operations during outbound processing. The value for this directory is noted by you on the Generate artifacts screen of the enterprise service discovery wizard.

Note: For the tutorials, you are encouraged to create this folder in a specific location. This path is noted in the tutorial introduction prerequisites.

The staging directory is used by the adapter to write the initial output files for create, append, and overwrite operations during outbound processing. The value for this directory is noted by you on the Generate artifacts screen of the enterprise service discovery wizard.

Note: For the tutorials, you are encouraged to create this folder in a specific location. This path is noted in the tutorial introduction prerequisites.

Configuring the adapter for outbound processing

To configure WebSphere Adapter for Flat Files for outbound processing, use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties for enterprise service discovery, select business objects or services that are in the enterprise information system, and generate business object definitions and related artifacts for outbound processing.

Generating business objects using enterprise service discovery

Use the enterprise services discovery wizard, which is launched from WebSphere Integration Developer, to generate business objects. Generating business objects involves the following tasks: setting the connection properties for the enterprise service discovery, selecting business objects or services that are in the enterprise information system, and generating artifacts for outbound processing.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to set the connection properties necessary to enable communication with the enterprise information file system. Once communication is established between the two entities, the enterprise service discovery wizard can obtain the necessary metadata from the enterprise information file system.

Before you begin

Before you can set the connection properties in this section, you must have created your adapter project. In addition, WebSphere Integration Developer must be running in order to complete this task.

About this task

Specify the connection properties that the enterprise service discovery wizard uses to communicate with the enterprise information system and discover its business objects and services.

To specify the connection properties for the enterprise service discovery wizard, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer window, switch to the Business Integration perspective.
 - a. Select **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed.
 - b. Select the **Business Integration** perspective.
2. Right-click the frame of the Business Integration perspective window and select **New** → **Enterprise Service Discovery**. If **Enterprise Service Discovery** is not visible, select **Other** from the bottom of the menu. Then, in the Select a wizard window, expand the **Business Integration** folder, select **Enterprise Service Discovery**, and click **Next**.
3. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for Flat Files (version 6.0.2) Connector Project** and click **Next**.
4. In the Configure Settings for Discovery Agent window, specify the properties used to discover the business data and the data binding used at run time. Data binding converts unstructured data into a business object. The properties specified in this task are optional and enable the data transformation framework functionality in the adapter. If the values of these properties are empty, the enterprise service discovery wizard treats creates a pass through scenario without data transformation.

To enable data transformation, use the following procedure.

- a. In the **Folder name** field, browse to the location where your XSD schemas for the business objects are stored.
- b. Optional: In the **Character Set** field, type the encoding the adapter uses to write files. By default, the value of this property is empty.

Note: If the contents of the file are in English, you do not need to select a Character Set. However, if the file content requires certain encoding, then you must select that particular value from the Character Set list in order for the adapter to use that particular encoding while writing to the enterprise information file system.

- c. From the **Content Type** list, select the format to be used for all business objects. This is a one time setting and is used to bundle a content type with a corresponding data binding.
- d. In the **DataBindingType** field, accept the default value, `XMLBOSerializerDataBinding`. This is the name of the data binding corresponding to the content type.

Note: The **DataBindingType** field is automatically populated based on the value selected for the **Content Type** field. If the value of the Content Type property is empty, the DataBindingType property does not display.

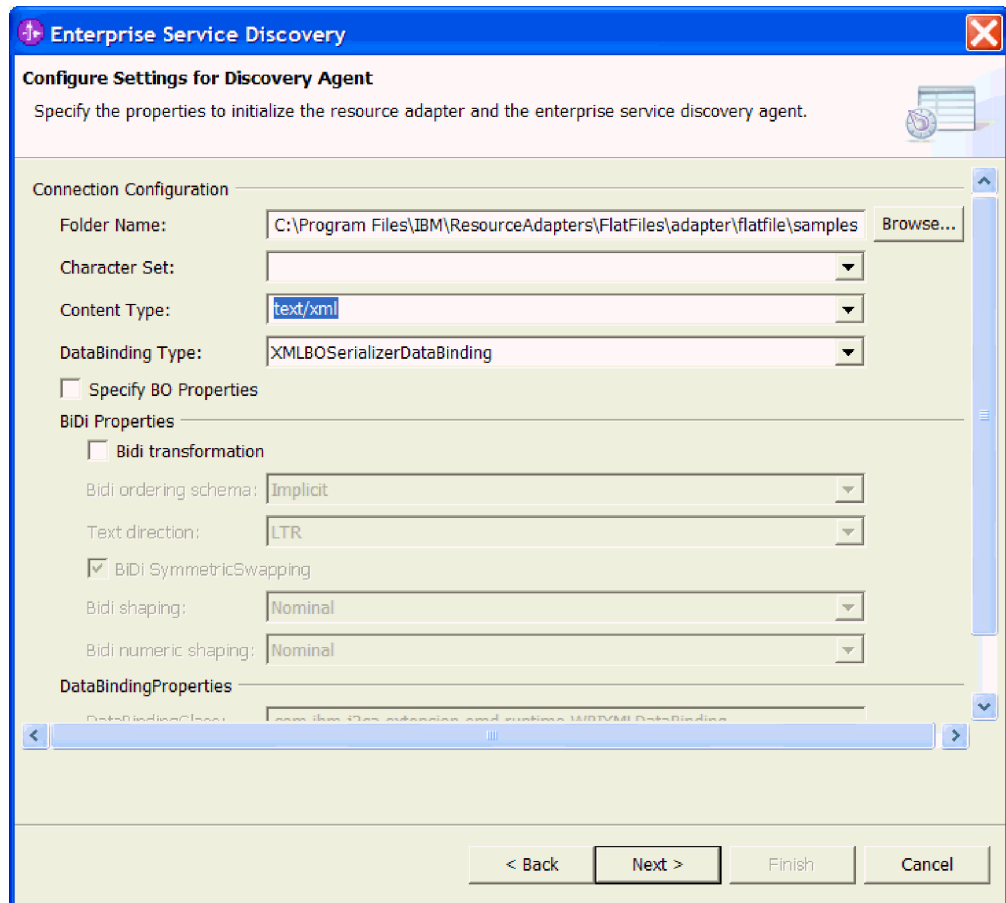


Figure 13. The Configure Settings for Discovery Agent window

5. Optional: To configure additional business object properties, select the **Specify BO Properties** check box. If this is selected, a window will display later in the Configure objects window. This is where individual business object properties can be defined.
6. Optional: To enable bidirectional transformation, select the **BiDi transformation** check box. Enabling this property automatically selects the **Select BiDi Symmetric Swapping** check box.

Note: If the enterprise information system supports a bidirectional format other than the Windows standard bidirectional format, then you will need to make appropriate changes to the properties below.

- a. From the **BiDi ordering schema** list, select the text orientation bidi parameter.
 - b. From the **Text direction** list, select the text direction bidi parameter.
 - c. From the **BiDi Shapping** list, select the Arabic shapping bidi parameter.
 - d. From the **BiDi numeric shapping** list, select the Arabic numeric shapping bidi parameter.
7. Optional: To set the logging level so that you can see errors during configuration, use the following procedure. Required fields are marked with an asterisk (*).
 - a. Click **Show Advanced**.
 - b. In the **Log file output location** field, type the path of the log file.

- c. From the **Logging Level** list, select the level of logging.
8. Click **Next**.

Result

The connection properties that the enterprise service discovery wizard uses to communicate with the enterprise information system are set.

What to do next

Use the enterprise service discovery wizard to view and select objects and services on the enterprise information file system for use in configuring the adapter.

Selecting business objects and services for use with the adapter

Use the enterprise service discovery wizard to browse the business objects and other metadata information on the enterprise information system and select artifacts to use to configure outbound processing.

Before you begin

Adapter connection properties must have been specified for the enterprise service discovery wizard before selecting the business objects or services from the enterprise information system to use in configuring the adapter.

About this task

Select the business objects or services that you will use to configure the adapter for outbound processing.

To select business objects and services for use with the adapter, use the following procedure.

How to perform this task

1. In the Find and Discover Enterprise Services window, click **Execute Query** to display the business objects for the adapter.
2. In the "Objects discovered by query" pane, highlight the business objects you want to import, then click **Add to import list** to move the business objects to the "Objects to be imported" pane.

Note: To remove objects from the "Objects to be imported" pane, highlight the object that you want to remove and click **Remove**.

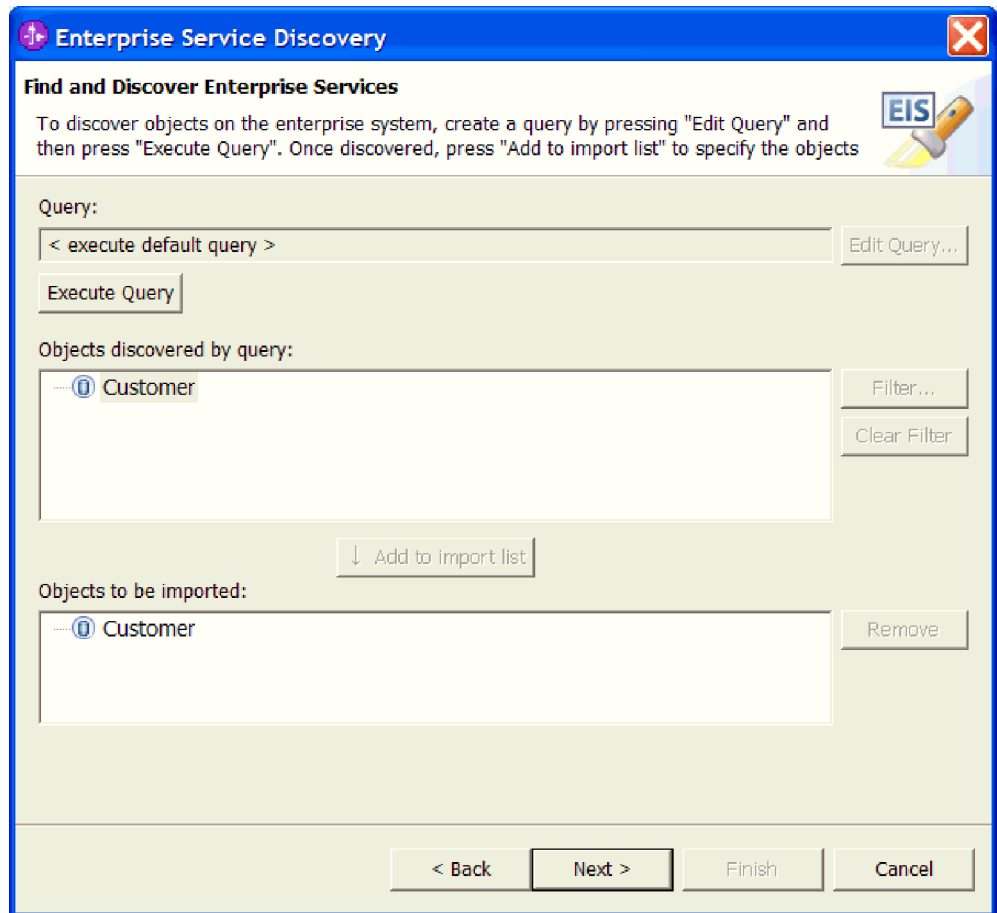


Figure 14. The Find and Discover Enterprise Services window

3. Click **Next**.
4. Optional: If you selected **Specify BO properties** in the Configure Settings for Discovery Agent window, the Configuration Parameters window displays for you to specify your individual business object properties.
 - a. From the **Character Set** list, select a character set. By default, the value of this property is empty.
 - b. From the **Content Type** list, select a content type. This is a one time setting and is used to bundle a content type with a corresponding data binding. It is the format used for all business objects. By default, the value of this property is empty.
 - c. Select the **Expose In Service Description** check box if the business object must be exposed in the service description as either input or output.
 - d. Click **OK**.
5. Click **Next**.

Configuring the selected objects

Once you have added the business object to the module, configure it for outbound operations.

1. In the Configure Objects window of the enterprise service discovery wizard, select **Outbound** from the **Service Type** list. The default base namespace for the business object schema to be generated is displayed. This value can be changed.

2. Type the location of the business object in the **BO Location** field. This creates the specified directory name in your connector project.
3. Click **Next**. All of the listed operations are selected by default. You can change the list by clicking the **Add** or **Remove** buttons.

The objects are now configured for outbound communication.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as a module.

Before you begin

To generate business objects and artifacts for your adapter project, you must have already selected business objects to add to the adapter project.

About this task

To generate business objects and artifacts for your adapter project, you must first create a new business integration module. Business integration modules contain all of the business processes that you will need for your business integration project. You will later deploy this module on the server.

To generate artifacts with the enterprise service discovery wizard, use the following procedure.

How to perform this task

1. To create a new business integration module, use the following procedure.
 - a. In the Generate Artifacts window, next to the **Module name** field, click **New**.
 - b. In the Integration Project window, select the default setting, **Create a module project**, and click **Next**.
 - c. In the New Module window, type the module name in the **Module Name** field.
 - d. Under Module Location, select the **Use Default** check box.
 - e. Click **Finish**.
2. In the Generate Artifacts window, select the default setting, **Use discovered connection properties** to set properties at this time. If you want to use the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console to configure the adapter and you do not want the import or export to contain the resource adapter's runtime properties, select **Use connection properties specified on server**.

Note: The adapter only supports the **Use discovered connection properties** option.

3. Optional: In the Generate Artifacts window, type the values for the managed connection factory properties. These properties are used by the adapter to create outbound connections to the enterprise information system file repository.
 - a. In the **Output Directory** field, type the path of the directory the adapter will use to write output files during outbound processing. This directory

must be created on your file system for this value to work. This value tells the adapter where it will write output files during outbound processing.

- b. In the **Staging Directory** field, type the path of the directory used by the adapter during create, append, and overwrite operations to first write output files to during outbound processing. After the file is written to the staging directory, the file is renamed and copied to the output directory.

Note: This directory must be created on your file system for this value to work. This value tells the adapter where it will create, append, and overwrite operations during outbound processing.

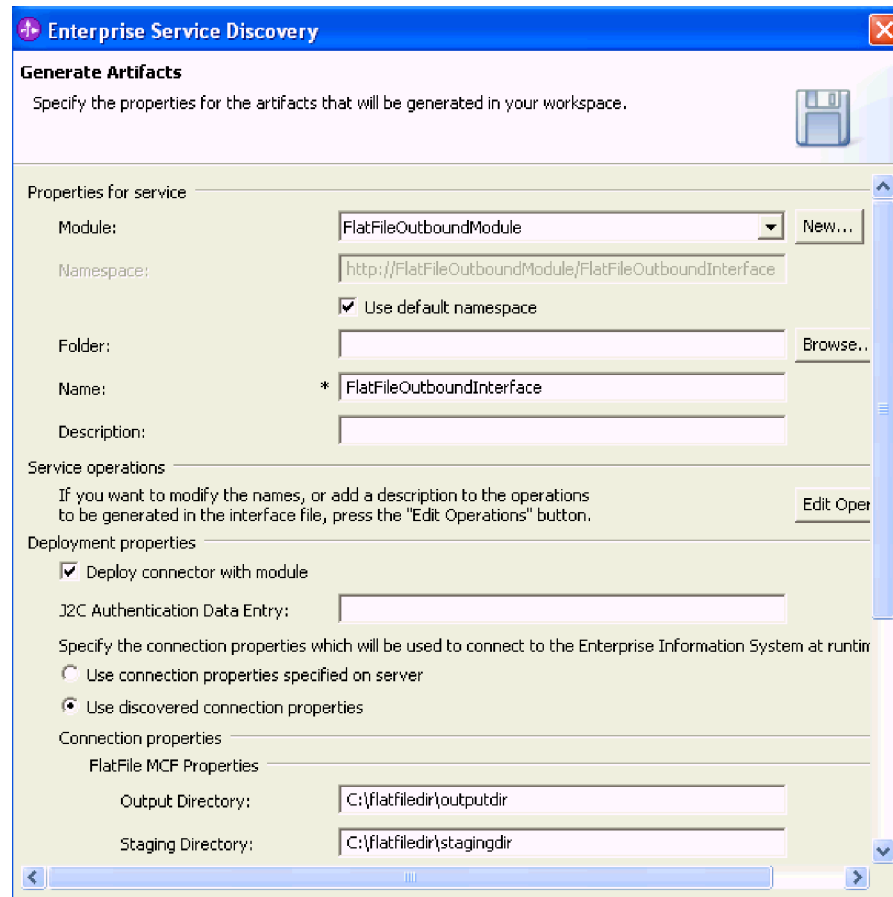


Figure 15. Generate Artifacts window

4. Type the value for each of the resource adapter properties. Required fields are marked with an asterisk (*).
 - a. In the **Adapter ID** field, type the identification of the adapter deployment instance.

Important: If you are using more than one instance of an adapter, the value of this property must be unique.

- b. In the **Log file size** field, type the size for each log file in kilobytes. If no value is specified, the file will have no maximum size.
- c. In the **Log file name** field, type the full path of the log file.

- d. In the **Log files** field, type the maximum number of log files to use after the log file has reached its maximum size. When a log file reaches its maximum size it will start using another log file. If no value is specified it will be set to 1.
- e. In the **Trace file size** field, type the size for each trace file in kilobytes. If no value is specified, the file will have no maximum size.
- f. In the **Trace file name** field, type the full path of the trace file.
- g. In the **Trace files** field, enter the maximum number of trace files to use after the log file has reached its maximum size. When a trace file reaches its maximum size it will start using another log file. If no value is specified it will be set to 1.

The screenshot shows the 'Connection properties' dialog box with the following settings:

- FlatFile MCF Properties**
 - OutputDirectory:
 - StagingDirectory:
- ResourceAdapterProperties**
 - Logging and Tracing**
 - Adapter ID [String]: *
 - Log file size [Integer]:
 - Log file name [String]:
 - Log Files [Integer]:
 - Trace file size [Integer]:
 - Trace file name [String]:
 - Trace files [Integer]:

Figure 16. Lower section of the Generate Artifacts window

5. Click **Finish**.

Result

The FlatFileOutboundInterface.wsdl, FlatFileOutboundInterface.import artifacts, FlatFileBG, FlatFile, UnstructuredContent, CustomerWrapperBG, CustomerWrapper, and Customer business objects are generated. The application business objects specified by the user are updated with application-specific information for data transformation and saved in the business object location.

What to do next

Use the enterprise service discovery wizard to configure the adapter for inbound processing.

Configuring the adapter for inbound processing

To configure WebSphere Adapter for Flat Files for inbound processing, use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties for the adapter, select business objects or services that are in the enterprise information system, and generate business object definitions and related artifacts for inbound processing.

Generating business objects using enterprise service discovery

To generate the business objects needed to communicate with the file system on the enterprise information system, you first set properties so that enterprise service discovery knows how to connect to the server. You then specify which services you want to use, and the required artifacts are generated by enterprise service discovery.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to set the connection properties necessary to enable communication with the enterprise information file system. Once communication is established between the two entities, the enterprise service discovery wizard can obtain the necessary metadata from the enterprise information file system.

Before you begin

Before you can set the connection properties in this section, you must have created your adapter project. In addition, WebSphere Integration Developer must be running in order to complete this task.

About this task

Specify the connection properties that the enterprise service discovery wizard uses to communicate with the enterprise information system and discover its business objects and services.

To specify the connection properties for the enterprise service discovery wizard, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer window, switch to the Business Integration perspective.
 - a. Select **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed.
 - b. Select the **Business Integration** perspective.
2. Right-click the frame of the Business Integration perspective window and select **New** → **Enterprise Service Discovery**. If **Enterprise Service Discovery** is not visible, select **Other** from the bottom of the menu. Then, in the Select a wizard window, expand the **Business Integration** folder, select **Enterprise Service Discovery**, and click **Next**.
3. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for Flat Files (version 6.0.2) Connector Project** and click **Next**.
4. In the Configure Settings for Discovery Agent window, specify the properties used to discover the business data and the data binding used at run time. Data

binding converts unstructured data into a business object. The properties specified in this task are optional and enable the data transformation framework functionality in the adapter. If the values of these properties are empty, the enterprise service discovery wizard treats creates a pass through scenario without data transformation.

To enable data transformation, use the following procedure.

- a. In the **Folder name** field, browse to the location where your XSD schemas for the business objects are stored.
- b. Optional: In the **Character Set** field, type the encoding the adapter uses to write files. By default, the value of this property is empty.

Note: If the contents of the file are in English, you do not need to select a Character Set. However, if the file content requires certain encoding, then you must select that particular value from the Character Set list in order for the adapter to use that particular encoding while writing to the enterprise information file system.

- c. From the **Content Type** list, select the format to be used for all business objects. This is a one time setting and is used to bundle a content type with a corresponding data binding.
- d. In the **DataBindingType** field, accept the default value, XMLBOSerializerDataBinding. This is the name of the data binding corresponding to the content type.

Note: The **DataBindingType** field is automatically populated based on the value selected for the **Content Type** field. If the value of the Content Type property is empty, the DataBindingType property does not display.

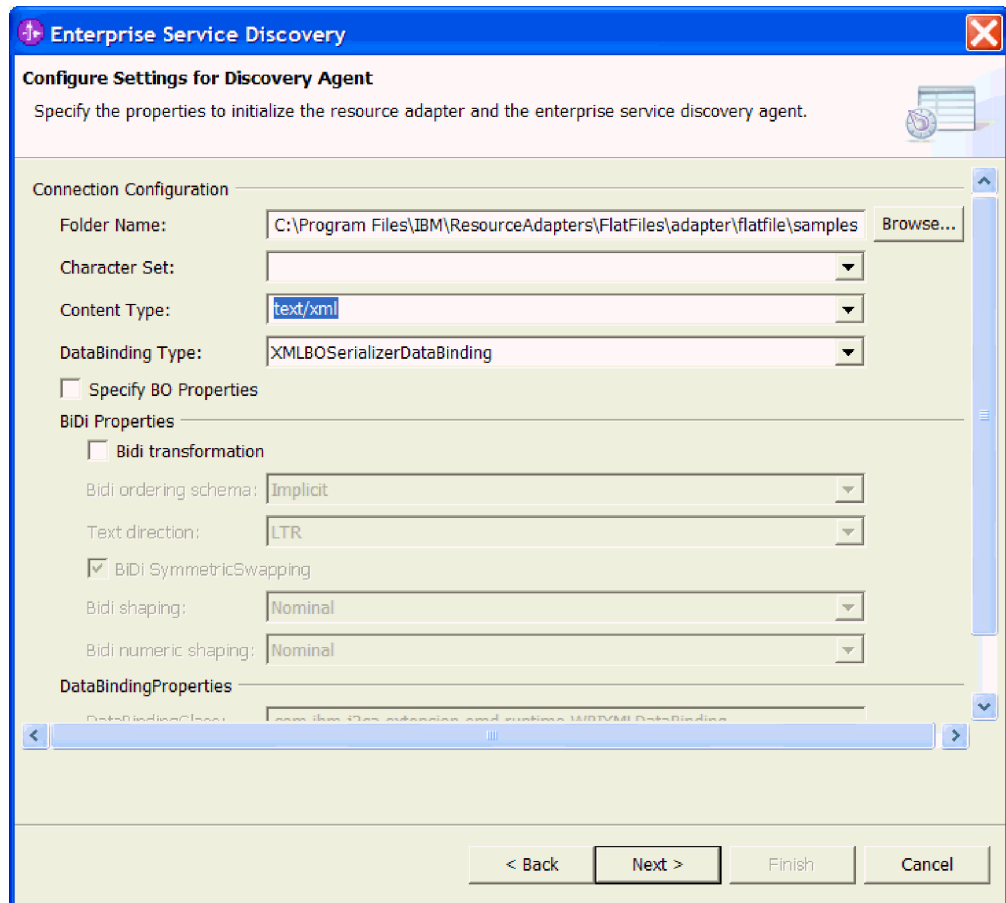


Figure 17. The Configure Settings for Discovery Agent window

5. Optional: To configure additional business object properties, select the **Specify BO Properties** check box. If this is selected, a window will display later in the Configure objects window. This is where individual business object properties can be defined.
6. Optional: To enable bidirectional transformation, select the **BiDi transformation** check box. Enabling this property automatically selects the **Select BiDi Symmetric Swapping** check box.

Note: If the enterprise information system supports a bidirectional format other than the Windows standard bidirectional format, then you will need to make appropriate changes to the properties below.

- a. From the **BiDi ordering schema** list, select the text orientation bidi parameter.
 - b. From the **Text direction** list, select the text direction bidi parameter.
 - c. From the **BiDi Shapping** list, select the Arabic shapping bidi parameter.
 - d. From the **BiDi numeric shapping** list, select the Arabic numeric shapping bidi parameter.
7. Optional: To set the logging level so that you can see errors during configuration, use the following procedure. Required fields are marked with an asterisk (*).
 - a. Click **Show Advanced**.
 - b. In the **Log file output location** field, type the path of the log file.

- c. From the **Logging Level** list, select the level of logging.
8. Click **Next**.

Result

The connection properties that the enterprise service discovery wizard uses to communicate with the enterprise information system are set.

What to do next

Use the enterprise service discovery wizard to view and select objects and services on the enterprise information file system for use in configuring the adapter.

Selecting business objects and services for use with the adapter

Use the enterprise service discovery wizard to browse the business objects and other metadata information on the enterprise information system and select artifacts to use to configure inbound processing.

Before you begin

You must have specified the adapter connection properties for the enterprise service discovery wizard to select the business objects or services from the enterprise information system to use in configuring the adapter.

About this task

After you select connection properties, select the business objects or services that you will use to configure the adapter for inbound processing.

To select business objects and services for use with the adapter, use the following procedure.

How to perform this task

1. In the Find and Discover Enterprise Services window, click **Execute Query** to display the business objects for the adapter.
2. In the "Objects discovered by query" pane, highlight the business objects you want to import, then click **Add to import list** to move the business objects to the "Objects to be imported" pane.

Note: To remove objects from the "Objects to be imported" pane, highlight the object that you want to remove and click **Remove**.

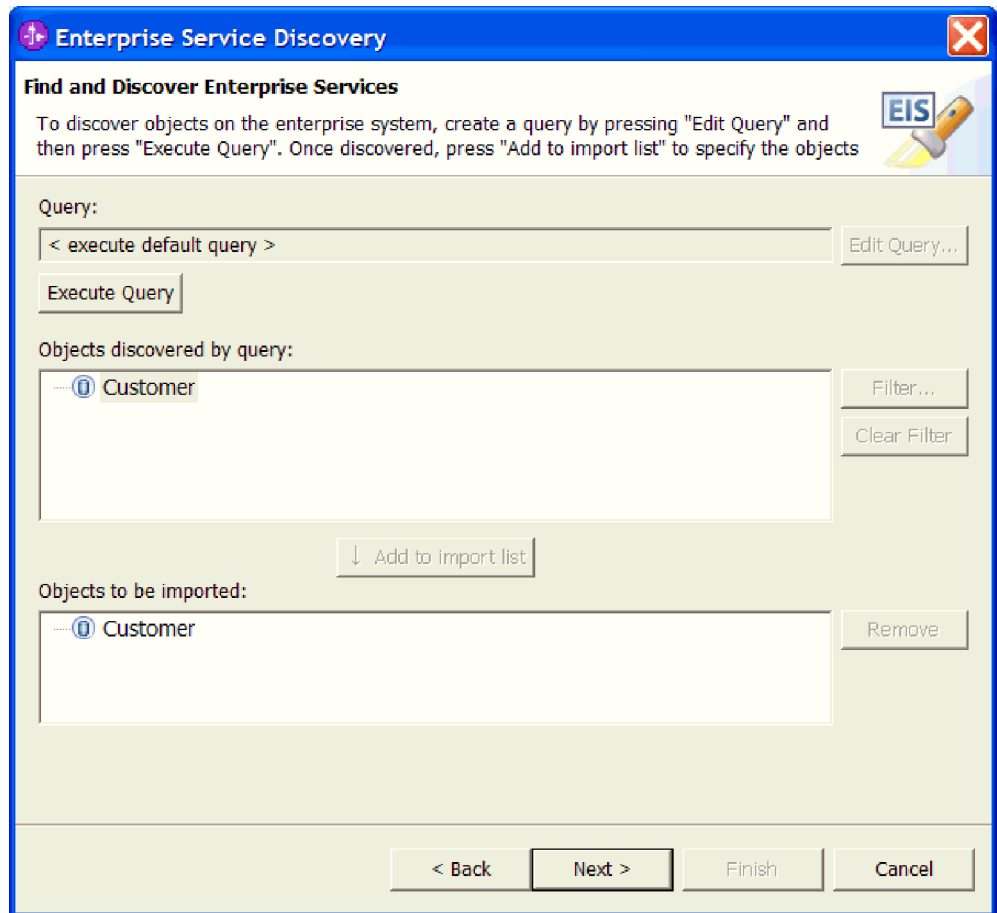


Figure 18. The Find and Discover Enterprise Services window

3. Click **Next**.
4. Optional: If you selected **Specify BO properties** in the Configure Settings for Discovery Agent window, the Configuration Parameters window displays for you to specify your individual business object properties.
 - a. From the **Character Set** list, select a character set. By default, the value of this property is empty.
 - b. from the **Content Type** list, select a content type . This is a one time setting and is used to bundle a content type with a corresponding data binding. It is the format used for all business objects. By default, the value of this property is empty.
 - c. Select the **Expose In Service Description** check box if the business object must be exposed in the service description as either input or output.
 - d. Click **OK**.
5. Click **Next**.

Configuring the selected objects

Once you have added business objects to the module, configure them for inbound operations.

1. In the Configure Objects window of the enterprise service discovery wizard, select **Inbound** from the **Service Type** list. The default base namespace for the business object schema to be generated is displayed. This value can be changed.

2. Type xsds in the **BO Location** field. This creates the specified directory to store your business objects called xsds in your connector project.
3. Click **Next**. All of the listed operations are selected by default. You can change the list by clicking the **Add** or **Remove** buttons.

Result

You have configured the objects that will be used with the inbound adapter project.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as a module.

Before you begin

To generate business objects and artifacts for your connector project, you must have already selected business objects to add to the adapter project. For more information on the values needed to complete the wizard fields, see the Reference section of this guide.

About this task

To generate business objects and artifacts for your connector project, you must first create a new business integration module. Business integration modules contain all of the business processes that you will need for your business integration project. You will later deploy this module on the server.

To generate artifacts with the enterprise service discovery wizard, use the following procedure.

How to perform this task

1. To create a new business integration module, use the following procedure.
 - a. In the Generate Artifacts window, next to the **Module name** field, click **New**.
 - b. In the Integration Project window, select the default setting, **Create a module project**, and click **Next**.
 - c. In the New Module window, type the module name in the **Module Name** field.
 - d. Under Module Location, select the **Use Default** check box.
 - e. Click **Finish**.
2. In the Generate Artifacts window, select the **Deploy connector with module** check box.
3. Select the default setting, **Use discovered connection properties**. If you want to use the WebSphere administrative console to configure the adapter and you do not want the import or export to contain the resource adapter's runtime properties, select **Use connection properties specified on server**.

Note: The adapter only supports the **Use discovered connection properties** option.

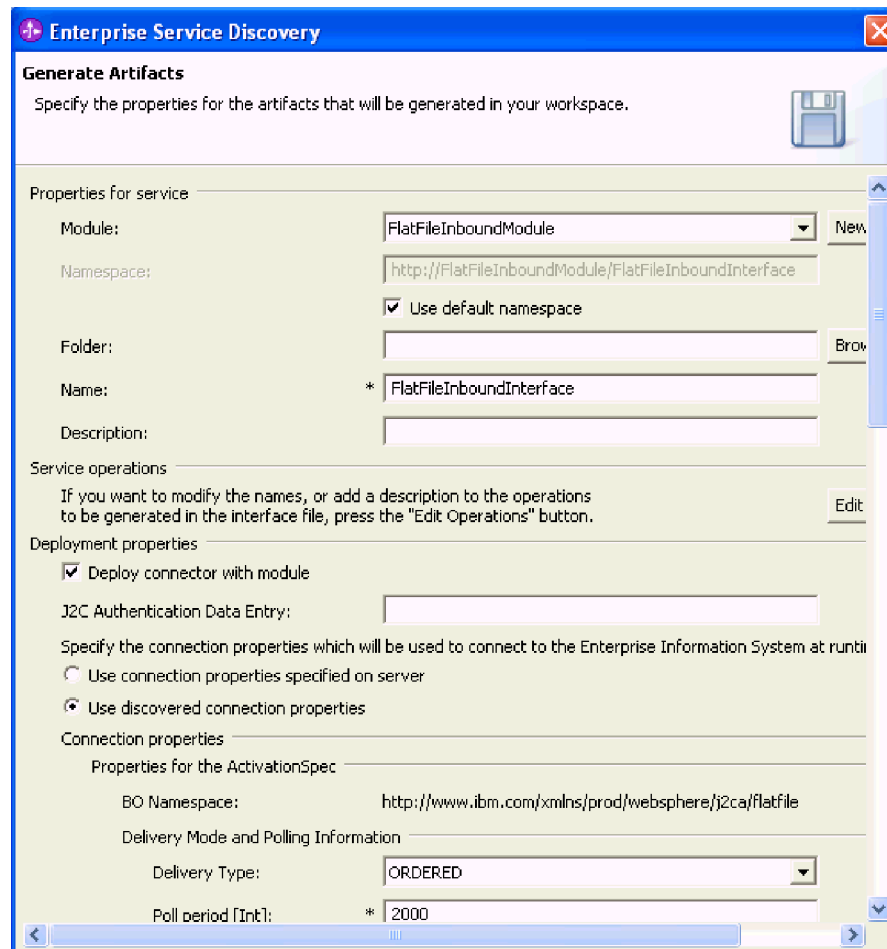


Figure 19. Generate Artifacts window

4. In the Generate Artifacts window, type the value for each of the required activation specification properties. Activation specification properties hold the inbound event processing configuration information for a message endpoint. Required fields are marked with an asterisk (*).
 - a. From the **Delivery type** list, select the delivery type for the endpoint. By default, ORDERED is selected.
 - b. In the **Poll period** field, type the time in milliseconds between polls for events. By default, the value of this property is 2000.
 - c. In the **Poll quantity** field, type the maximum number of events to be collected during each poll. By default, the value of this property is 10.
 - d. In the **Retry interval** field, type the time in milliseconds between connection attempts in case of enterprise information system connection failure.
 - e. In the **Retry limit** field, type the number of times the adapter attempts to establish an inbound connection.
 - f. To stop the adapter when an error is encountered during polling, select the **Stop polling on error** check box.
 - g. To disable once-and-only-once delivery, clear the **Assured once delivery** check box. By default, this property is selected.
 - h. To filter event files that have a timestamp, select the **Filter future events** check box.

- i. In the **Event type filter** field, type the business object types that the adapter should publish.
5. In the Generate Artifacts window, type a value for each of the required Flat Files properties. These are activation specification properties that are specific to the adapter. Required fields are marked with an asterisk (*).
- a. In the **EventDirectory** field, type the repository path name where the event files are stored. The event directory must be created on your local directory or the wizard will not be able to find inbound events.
 - b. To enable archiving, select the **Archiving processed** check box
 - c. If archiving is enabled, then In the **Archive directory** field, type the path name where processed event files are archived.
 - d. In the **FailedArchiveExt** field, type the file extension used to archive unsuccessful business objects. By default, the value of this property is fail.
 - e. In the **OriginalArchiveExt** field, type the file extension used to archive the original event file. By default, the value of this property is original.
 - f. In the **SuccessArchiveExt**, type the file extension used to archive all successful business objects. By default, the value of this property is success.
 - g. In the **EventFileMask** field, type the value of the file mask used while the adapter polls the event directory. By default, the value of this property is `**.*`.
 - h. To pass an unprocessed event file to the endpoint as a reference, select the **FilePassByReference** check box.
 - i. To send the value of the **SplitCriteria** property to the endpoint along with the contents of the business object, select the **IncludeEndBODelimiter** check box.
 - j. To manually the event persistence database, clear the **EPCreateTable** check box. By default, the value of this property is true.
 - k. In the **DataSourceJNDIName** field, type the JNDI name of the data source created on WebSphere Process Server. This property is used by event persistence to obtain the JDBC database connection.
 - l. In the **EventTableName** field, type the name of the table used by the adapter for event persistence.
 - m. In the **DatabaseSchemaName** field, type the schema name of the database used for event persistence.
 - n. In the **DatabaseUserName** field, type the user name used by event persistence to obtain the JDBC database connection from the data source.
 - o. In the **DatabasePassword** field, type the password used by event persistence to obtain the JDBC database connection from the data source.
 - p. In the **SortEventFiles** field, type a value that determines the sorting order of polled event files.
 - q. In the **SplitCriteria** field, a value to split event files based on either size or a delimiter.
 - r. In the **SplittingFunctionClassName** field, type the name of the class used to split files based on either size or a delimiter.
 - s. In the **EventContentType** field, type the mime type of the event file.
 - t. In the **FileContentEncoding** field, type the encoding used for the content of the event file.
 - u. In the **DefaultObjectName** field, type the namespace of the wrapper business graph used during inbound processing.

6. In the Generate Artifacts window, type the value for each of the resource adapter properties. Required fields are marked with an asterisk (*).
 - a. In the **Adapter ID** field, type the identification of the adapter deployment instance.

Important: If you are using more than one instance of an adapter, the value of this property must be unique.

- b. In the **Log file size** field, type the size for each log file in kilobytes. If no value is specified, the file will have no maximum size.
 - c. In the **Log file name** field, type the full path of the log file.
 - d. In the **Log files** field, type the maximum number of log files to use after the log file has reached its maximum size. When a log file reaches its maximum size it will start using another log file. If no value is specified it will be set to 1.
 - e. In the **Trace file size** field, type the size for each trace file in kilobytes. If no value is specified, the file will have no maximum size.
 - f. In the **Trace file name** field, type the full path of the trace file.
 - g. In the **Trace files** field, enter the maximum number of trace files to use after the log file has reached its maximum size. When a trace file reaches its maximum size it will start using another log file. If no value is specified it will be set to 1.
7. Click **Finish**.

Result

The FlatFileInboundInterface.wsdl and FlatFileInboundInterface.export artifacts, and the FlatFileBG, FlatFile, UnstructuredContent, CustomerWrapperBG, CustomerWrapper and Customer business objects are generated. The application business objects specified by the user are updated with application-specific information for data transformation and saved in the business object location.

What to do next

Use the WebSphere Integration Developer assembly editor to generate reference bindings.

Generating reference bindings

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

Before you begin

An adapter project must be created and configured on your workspace. In addition, WebSphere Integration Developer must be running in order to complete this task.

About this task

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.

To generate reference bindings, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer window, switch to the Business Integration perspective.
 - a. Select **Window** → **Open Perspective** → **Other**.
 - b. Select the **Business Integration** from the list of perspectives that are displayed.
2. In the Business Integration Perspective of WebSphere Integration Developer, right-click the inbound module, and select **Open With** → **Assembly Editor**. The Assembly Diagram window appears with the modules Import component in view.

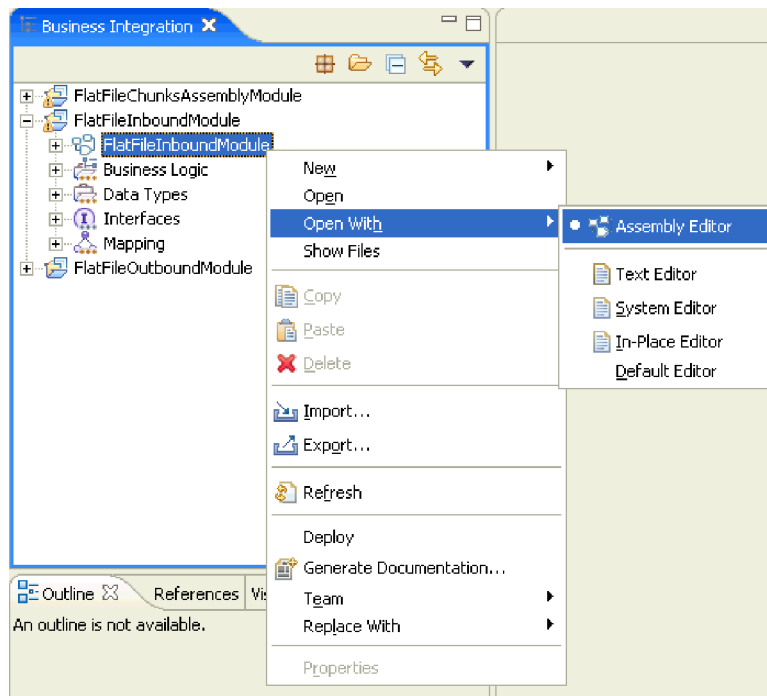


Figure 20. Open Assembly Editor

3. Click the **Component (with no implementation type)** icon.

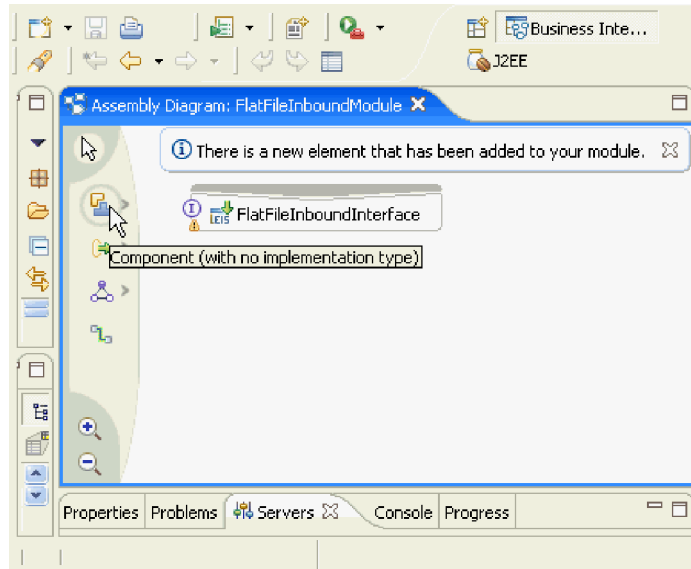


Figure 21. The Component icon in the Assembly Diagram window

A new menu of icons appears.

4. Click the **Component (with no implementation type)** icon and drag the component selection to the editor workspace.

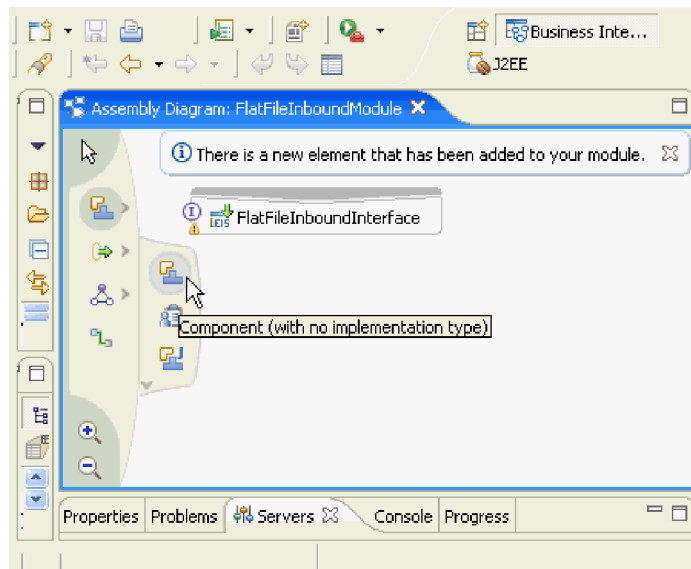


Figure 22. Second component icon

5. Click the **Wire** icon and drag the wire from **FlatFileInboundInterface** to **Component1** to create the wiring.

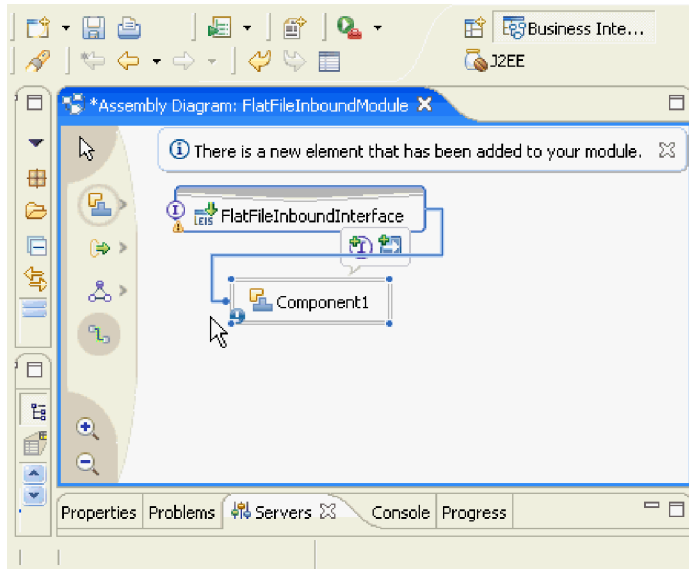


Figure 23. Component wiring window

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 act like an endpoint listener. During event delivery of the inbound operation, the adapter invokes the emitFlatFile 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 when prompted and click **OK**.
 - d. Click **OK** in the Generate Implementation window.
 - e. In the generated Java implementation, scroll to the emitFlatFile method.
 - f. Add custom code inside the emitFlatFile method to process the delivered business object according to your specific business needs.
8. Click **File** → **Save**.

Result

You have created a reference binding for your adapter project.

What to do next

Use WebSphere Integration Developer to test the assembled adapter package.

Chapter 7. Deploying the module

To deploy the module to the application server, export the adapter project as an enterprise archive (EAR) file, install the module, and add any configuration properties that were not set in the enterprise service discovery wizard.

Exporting the project as an enterprise archive resource (EAR) file

Using the enterprise service discovery wizard, export the adapter project that you have created as an EAR file. By creating an EAR file, you capture all of the contents of your adapter project in a format can be easily deployed to the application server.

Before you begin

Before you can export the project as an EAR file, you must have created your business objects and generated reference bindings.

About this task

To export the project as an EAR file, perform the following procedure.

How to perform this task

1. From WebSphere Integration Developer, switch to the J2EE perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **J2EE** and click **OK**.
2. In the Project Explorer pane, expand **Enterprise Applications**.
3. Right-click the assembled adapter module, then click **Export** → **EAR file**.
4. In the EAR Export window, enter the following information:
 - a. From the **EAR project** list, select the name of the assembled adapter application.
 - b. From the **Destination** list, either select or browse to the exact location of the EAR file.
 - c. Select the **Export source files** check box.
 - d. Select the **Overwrite existing file** check box.
 - e. Select the **Include project build paths and meta-data files** check box.
 - f. Click **Finish**.

Result

The assembled adapter project, now known as an SCA module that contains an EIS import, is saved in the destination folder and ready to be installed on the server.

What to do next

Use the server administrative console to install the application on WebSphere Process Server or WebSphere Enterprise Service Bus.

Installing the module

Installing the adapter project is the last step of the deployment process. When you install the adapter project on the server and run it, the adapter, which is embedded as part of the project module, runs as part of the installed application.

Before you begin

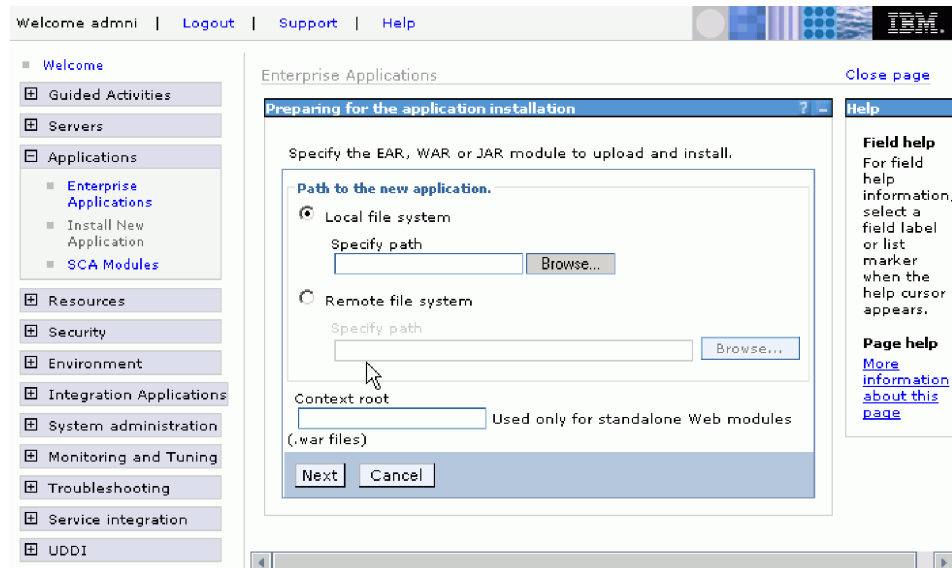
You must have exported your project module as an EAR file before installing the adapter project.

About this task

To install the adapter module, perform the following procedure. For more information on clustering adapter project applications, see <http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp>.

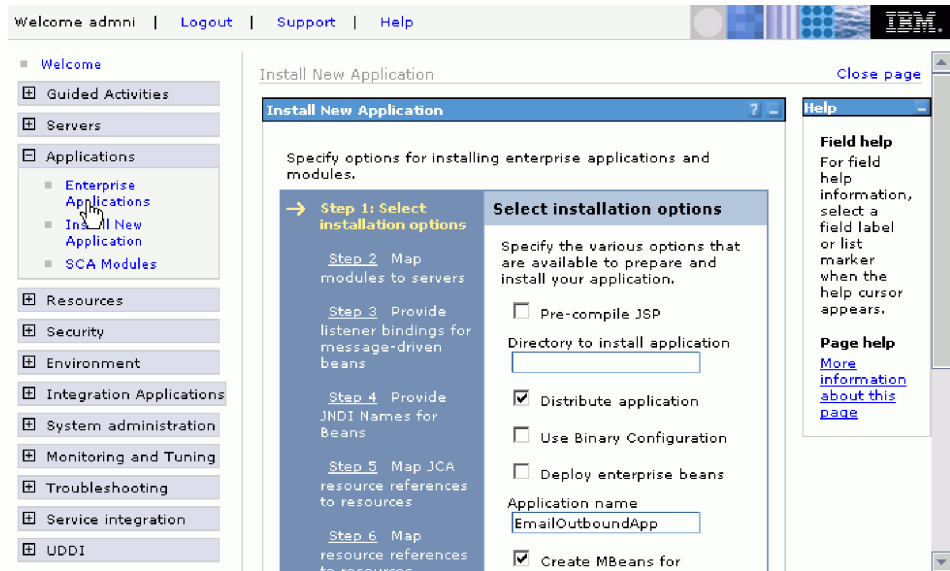
How to perform this task

1. Open the WebSphere Process Server administrative console by right-clicking your server instance and selecting **Run administrative console**.
2. In the administrative console window, click **Applications** → **Install New Applications**.



Preparing for the application installation window

3. Click **Browse** to locate your EAR file and click **Next**.
4. Optional: If you are deploying to a clustered environment, click **Next** until you reach Step 2: Mapping modules to servers, then select **Modules** and then the name of the server cluster and click **Apply**. Note: Adapter instances are replicated in a clustered server environment when `enableHASupport` is set to true. Do not change the value of `enableHASupport` for single server environments. **Note:** Adapter instances are replicated in a clustered server environment when `enableHASupport` is set to true. Do not change the value of `enableHASupport` for single server environments.
5. Click **Next** until you reach Step 6: Map resource reference to resources.



Install New Application window

6. Select **SCA Auth Alias** from the select authentication data entry list.
7. Select the check box for the module and click **Apply**.
8. Click **Next**. A summary of all of the installation options is displayed.
9. Verify that all options are correct and click **Finish**.
10. Confirm that the application was installed successfully.
11. Click the **Save to Master Configuration** link at the end of the list of installation messages.
12. Click **Save**.

Result

The project is now deployed and the Enterprise Applications window for the deployed application is displayed.

What to do next

If you want to set or reset resource adapter, managed connection factory, activation specification, or data transformation properties, or you would like to cluster adapter project applications, you should do that using the WebSphere Process Server administrative console before configuring troubleshooting tools.

Setting or changing adapter configuration properties

Adapter configuration properties are set using the enterprise service discovery wizard when you generate business objects. However, after you deploy the Adapter for Flat Files use the administrative console to update managed (J2C) connection factory properties (which are used for outbound processing) and activation specification properties (which are used for inbound processing).

Setting resource adapter properties

Resource adapter properties include logging and tracing properties, bidirectional properties, and adapter-specific properties. Using the enterprise service discovery wizard when you first configure the adapter (and later, via the administrative console), you can configure Resource Adapter properties.

Before you begin

Your adapter module must be deployed on the WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

Resource adapter properties are set using the enterprise service discovery wizard when you create your business objects. You can reconfigure the resource adapter properties using administrative console.

To configure properties using the administrative console, use the following procedure.

How to perform this task

1. Start the administrative console.
2. From the administrative console, expand **Resources**.
3. Expand **Resource Adapters** and select **WebSphere Adapter for Flat Files**.
4. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
5. Scroll to the bottom of the window. Under Related Items, click **Connector Modules**.
6. Click the **CYWFF_FlatFile.rar** file.
7. Click **Resource Adapter**.
8. Click **Custom properties**.
9. For each property you want to change, perform the following steps:
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field value or type a value, if the field is empty.
 - c. Click **OK**.
10. Click the **Save** link in the Messages box at the top of the window.

Result

You have reconfigured your Resource adapter properties using the administrative console.

What to do next

Use the administrative console to set or change managed connection factory properties.

Setting managed (J2C) connection factory properties

Managed (J2C) connection factory properties affect outbound processing and correspond to the ManagedConnectionFactory interface of the J2EE Connector Architecture Specification. A J2C connection factory also manages connection pooling. It provides configuration information for outbound operations via the resource adapter. Managed (J2C) connection factory properties are set using the enterprise service discovery wizard when you create your business objects. You can reconfigure the Managed (J2C) connection factory properties using the WebSphere Process Server administrative console.

Before you begin

Your adapter module must be deployed on the WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

Managed (J2C) connection factory properties are set using the enterprise service discovery wizard when you create your business objects. You can reconfigure the managed (J2C) connection factory properties using the WebSphere Process Server administrative console.

To configure properties using the administrative console, use the following procedure.

How to perform this task

1. Start the administrative console.
2. From the administrative console, expand **Resources**.
3. Expand **Resource Adapters** and select **WebSphere Adapter for Flat Files**.
4. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
5. Scroll to the bottom of the window. Under Related Items, click **Connector Modules**.
6. Under Additional Properties, select **J2C connection factories**.
7. Click the **CYWFF_FlatFile.rar** file.
8. Click **Resource Adapter**.
9. Click **J2C connection factories**.
10. For each property you want to change, perform the following steps:
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field value or type a value, if the field is empty.
 - c. Click **OK**.
11. Click the **Save** link in the Messages box at the top of the window.

Result

You have reconfigured your Resource adapter properties using the WebSphere Process Server administrative console.

What to do next

Set activation specification properties with the enterprise service discovery wizard.

Setting activation specification properties for the EIS

ActivationSpecification properties hold the inbound event processing configuration information for a message endpoint. You can reconfigure the activation specification properties using the administrative console.

Before you begin

Your adapter module must be deployed on the WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

ActivationSpecification properties are set using the enterprise service discovery wizard when you create your business objects. You can reconfigure the managed ActivationSpecification properties using the administrative console.

To configure properties using the administrative console, use the following procedure.

How to perform this task

1. Start the administrative console.
2. From the administrative console, expand **Resources**.
3. Expand **Resource Adapters** and select **WebSphere Adapter for Flat Files**.
4. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
5. Scroll to the bottom of the window. Under Related Items, click **Connector Modules**.
6. Click the **CYWFF_FlatFile.rar** file.
7. Click **J2C Activation specifications**.
8. Click **Custom properties**.
9. For each property you want to change, perform the following steps:
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field value or type a value, if the field is empty.
 - c. Click **OK**.
10. Click the **Save** link in the Messages box at the top of the window.

Result

You have reconfigured your ActivationSpecification properties using the WebSphere Process Server administrative console.

What to do next

Configure troubleshooting tools.

Chapter 8. Configuring troubleshooting tools

Configure the troubleshooting tools to suit your requirements. Enable logging for the adapter to control the status of event processing. Enable the Common Event Infrastructure to collect diagnostic information about your adapter. Set tracing levels to determine the level of the information captured in the adapter log and trace files. Install IBM Support Assistant to gain quick access to support-related information along with serviceability tools for problem determination for IBM software products.

Enabling tracing with the Common Event Infrastructure (CEI)

Enable tracing and control the level of detail in the adapter trace by configuring the Common Event Infrastructure (CEI).

Before you begin

Before you enable tracing with CEI, complete the following tasks:

- Enable the diagnostic trace service.
- Publish the IBM WebSphere Adapters event definitions file to the CEI catalog before you can set these event definitions.

For instruction on how to do these tasks, refer to the CEI documentation located on the Web site for your server:

- For WebSphere Process Server: <http://www.ibm.com/software/integration/wps>
- For WebSphere Enterprise Service Bus: <http://www.ibm.com/software/integration/wsesb>

To enable tracing and control the level of trace detail, use the following procedure.

How to perform this task

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. In the list of servers, click the name of your server.
4. In the General Properties area, click **Change Log Detail Level** and then select **com.ibm.j2ca.*** for the adapter components. There is a subcomponent for each adapter type, as described in the following table.

Adapter	Package Name
WebSphere Adapter for Email	com.ibm.j2ca.email.*
WebSphere Adapter for Flat Files	com.ibm.j2ca.flatfile.*
WebSphere Adapter for FTP	com.ibm.j2ca.ftp.*
WebSphere Adapter for JDBC	com.ibm.j2ca.jdbc.*
WebSphere Adapter for JD Edwards EnterpriseOne	com.ibm.j2ca.jde.*
WebSphere Adapter for SAP Software	com.ibm.j2ca.sap.*
WebSphere Adapter for Siebel Business Applications	com.ibm.j2ca.siebel.*

5. 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.adapter_ID1*. For each instance of a deployed adapter, the system shows a separate ID.

6. Select the CEI adapter ID that you want to enable.
7. From the list, choose the level of business object detail to capture in service component events:
 - **off**. Turn CEI off.
 - **fine**. Turn CEI on but publish none of the business object payload. This corresponds to the event control detail level of Empty in WebSphere Integration Developer.
 - **finer**. Turn CEI on and publish only the payload description for the business object. This corresponds to the event control detail level of Digest in WebSphere Integration Developer .
 - **finest**. Turn CEI on and publish all of the business object payload. This corresponds to the event control detail level of Full in WebSphere Integration Developer.
 - **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 for your process server.

Configuring logging properties

Use the administrative console to enable logging and to set the output properties for a log, including the location, level of detail, and output format of the log.

About this task

Before the adapters can log monitored events, you must specify the service component event points that you want to monitor, what level of detail you require for each event, and format of the output used to publish the events to the logs.

Use the administrative console to perform the following tasks:

- Enable or disable a particular event log
- Specify the level of detail in a log
- Specify where log files are stored and how many log files are kept
- Specify the format for log output

If you set the output for log analyzer format, you can open trace output using the Log Analyzer tool, which is an application included with your process server. This is useful if you are trying to correlate traces from two different server processes, because it allows you to use the merge capability of the Log Analyzer.

For more information about monitoring on a process server, including service components and event points, see the documentation for your process server.

You can change the log configuration statically or dynamically. Static configuration take effect when you start or restart the application server. Dynamic, or runtime, configuration changes apply immediately.

When a log is created, the detail level for that log is set from the configuration data. If no configuration data is available for a particular log name, the level for that log is obtained from the parent of the log. If no configuration data exists for the parent log, the parent of that log is checked, and so on up the tree, until a log with a non-null level value is found. When you change the level of a log, the change is propagated to the children of the log, which recursively propagate the change to their children, as necessary.

To enable logging and set the output properties for a log, use the following procedure.

How to perform this task

1. In the navigation pane of the administrative console, click **Servers** → **Application Servers**.
2. Click the name of the server that you want to work with.
3. Under **Troubleshooting**, click **Logs and trace**.
4. Click **Change Log Detail Levels**.
5. Specify when you want the change to take effect:
 - For a static change to the configuration, click the **Configuration** tab.
 - For a dynamic change to the configuration, click the **Runtime** tab.
6. Select the packages whose logging level you want to modify. The package names for WebSphere Adapters start with **com.ibm.j2ca**:
 - For the adapter base component, select **com.ibm.j2ca.base**.
 - For the adapter base component and all deployed adapters, select **com.ibm.j2ca.base.***.
 - For a specific adapter, select its package name.

Adapter	Package Name
WebSphere Adapter for Email	com.ibm.j2ca.email
WebSphere Adapter for Flat Files	com.ibm.j2ca.flatfile
WebSphere Adapter for FTP	com.ibm.j2ca.ftp
WebSphere Adapter for JDBC	com.ibm.j2ca.jdbc
WebSphere Adapter for JD Edwards EnterpriseOne	com.ibm.j2ca.jde
WebSphere Adapter for SAP Software	com.ibm.j2ca.sap
WebSphere Adapter for Siebel Business Applications	com.ibm.j2ca.siebel

7. Click the package name and select the logging level.

Logging Level	Description
Fatal	The task cannot continue or the component cannot function.
Severe	The task cannot continue, but the component can still function. This logging level also includes conditions that indicate an impending fatal error, that is, situations that strongly suggest that resources are on the verge of being depleted.
Warning	A potential error has occurred or a severe error is impending. This logging level also includes conditions that indicate a progressive failure, for example, the potential leaking of resources.
Audit	A significant event has occurred that affects the server state or resources.

Logging Level	Description
Info	The task is running. This logging level includes general information outlining the overall progress of a task.
Config	The status of a configuration is reported or a configuration change has occurred.
Detail	The subtask is running. This logging level includes general information detailing the progress of a subtask.

8. Click **Apply**.
9. Click **OK**.
10. To have static configuration changes take effect, stop and then restart the process server.

Changing the log and trace file names

By default, log and trace information for all processes and applications on a process server is written to the SystemOut.log and trace.log files, respectively. To keep the adapter log and trace information separate from other processes, use the administrative console to change the file names.

About this task

You can change the log and trace file names at any time after the adapter module has been deployed to an application server.

You can change the log configuration statically or dynamically. Static configuration changes affect applications when you start or restart the application server. Dynamic or run time configuration changes apply immediately.

Log and trace files are in the *install_root/profiles/profile_name/logs/server_name* folder.

To set or change the log and trace file names, use the following procedure.

How to perform this task

1. In the navigation pane, click **Enterprise Applications**.
2. Click the name of the adapter application. This is the name of the EAR file for the adapter, without the .ear file extension. For example, if the EAR file is named Accounting_OutboundApp.ear, then click **Accounting_OutboundApp**.
3. Click **Connector Modules**.
4. Select the adapter by clicking the name of the RAR file for the adapter. The RAR files are listed in the following table.

Adapter	RAR File Name
WebSphere Adapter for Email	CWYEM_Email.rar
WebSphere Adapter for Flat Files	WYFF_FlatFile.rar
WebSphere Adapter for FTP	CWYFT_FTPFile.rar
WebSphere Adapter for JDBC	CWYBC_JDBC.rar
WebSphere Adapter for JD Edwards EnterpriseOne	CWYED_JDE.rar
WebSphere Adapter for SAP Applications	CWYAP_SAPAdapter.rar CWYAP_SAPAdapterTX.rar

Adapter	RAR File Name
WebSphere Adapter for Siebel Business Applications	CWYEM_Siebel.rar

5. Click the name of the resource adapter.
6. In the Custom Properties area, specify the file names:
 - To change the log file name, type the name in the **Value** field for **logFilename**. By default, this log is in the SystemOut.log file.
 - To change the trace file name, type the name in the **Value** field for **traceFilename**. By default, this log is in the trace.log file.
7. To have static configuration changes take effect, stop and then restart the process server.

Installing or upgrading IBM Support Assistant

IBM Support Assistant (ISA) is a free, local software serviceability workbench that helps you resolve questions and problems with IBM software products. Install plug-ins for the products you have installed. It provides quick access to support-related information along with serviceability tools for problem determination. Installing and upgrading it is simple and straightforward.

About this task

IBM Support Assistant provides the following services:

- Symptom-based data collection
- Access to IBM support information, IBM newsgroups, and other resources through a federated search interface (one search, multiple resources)
- Easy access to IBM educational materials
- Easy access to IBM product home pages, product support pages, and product forums or newsgroups through convenient links
- A tools framework and update manager to easily update and install ISA plug-ins and tools
- Fast resolution of problem management records through electronic submission of critical system data to IBM

You can install and run both version 2 and version 3 of IBM Support Assistant on a single computer, to get support for a broad range of IBM solutions.

To install and upgrade IBM Support Assistant, use the following procedure.

How to perform this task

1. Go to the IBM Support Assistant Web page at:
<http://www.ibm.com/software/support/isa/>
2. Follow the directions on the Web page to download ISA version 3.0, and then to extract, install, and use the tool.
3. Start ISA.
4. Open the **Updater** component.
5. On the **Upgrades** tab, upgrade ISA to version 3.0.1 or higher.
6. On the **New Products and Tools** tab, install the plug-ins for your adapter. Select the plug-in for your adapter from the list for the WebSphere brand. There is an optional language pack plug-in for each adapter, which enables you to see adapter-specific information in languages other than English.

Chapter 9. Administering the adapter

Use the administrative console of the server to start, stop, and troubleshoot the adapter.

Starting the adapter

To start an adapter that has a status of Stopped, use the administrative console. By default, an adapter starts automatically when the server starts.

Before you begin

The administrative console of the server must be running in order to complete this task.

To start the adapter, use the following procedure.

How to perform this task

1. On the Enterprise Applications page, click **Applications** → **Enterprise Applications**.
2. Select the check box of the adapter that you want to start.
3. Click **Start**.

Result

The status of the adapter changes to Started and a message stating that the adapter started displays at the top the page.

Use the administrative console of the server to stop the adapter.

Stopping the adapter

Use the administrative console of the server to stop an adapter.

Before you begin

The administrative console of the server must be running in order to complete this task.

To stop the adapter, use the following procedure.

How to perform this task

1. On the Enterprise Applications page, click **Applications** → **Enterprise Applications**.
2. Clear the check box of the adapter you want to stop.
3. Click **Stop**.

Result

The status of the adapter changes to Stopped and a message stating that the adapter stopped displays at the top the page.

Use the administrative console of the server to troubleshoot the adapter.

Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly. If necessary, follow the procedures for contacting IBM Software Support.

Exception: XAResourceNotAvailableException

When the process server log contains repeated reports of the `com.ibm.ws.Transaction.XAResourceNotAvailableException` exception, remove transaction logs to correct the problem.

Symptom:

When the adapter starts, the following exception is repeatedly logged in the process server log file:

```
com.ibm.ws.Transaction.XAResourceNotAvailableException
```

Problem:

A resource was removed while the process server was committing or rolling back a transaction for that resource. When the adapter starts, it tries to recover the transaction but cannot because the resource was removed.

Solution:

To correct this problem, use the following procedure:

1. Stop the process server.
2. Delete the transaction log file that contains the transaction. Use the information in the exception trace to identify the transaction. This prevents the server from trying to recover those transactions.

Note: In a test or development environment, you can generally delete all of the transaction logs. In WebSphere Integration Developer, delete the files and subdirectories of the transaction log directory, `server_install_directory\profiles\profile_name\tranlog`.

In a production environment, delete only the transactions that represent events that you do not need to process. One way to do this is to reinstall the adapter, pointing it to the original event database used, and deleting only the transactions you do not need. Another approach is to delete the transactions from either the log1 or log2 file in the following directory:

```
server_install_directory\profiles\profile_name\tranlog\node_name\wps\  
server_name\transaction\tranlog
```

3. Start the process server.

Self help resources

Use the self help resources of IBM Software Support to get the most current support information, to obtain technical documentation, to download support tools and fixes, and prevent problems with WebSphere Adapter for Flat Files. The self help resources also help you diagnose problems with the adapter and contact IBM Software Support.

The software support Web site for WebSphere Adapters at <http://www.ibm.com/software/integration/wbiadapters/support> provides the following resources:

- Flashes (alerts from technical support)
- Technotes
You can get a list of technotes for WebSphere Adapters at <http://www.ibm.com/support/search.wss?rs=695&tc=SSMKUK>
- Authorized program analysis reports (APARs)
- Technical information including the product information center, manuals, IBM Redbooks™, and whitepapers.
- Educational offerings
- *IBM Software Support Handbook*

Register at the site to use My Support to create a customized support page for your use.

Contacting IBM Software Support

IBM Software Support provides support for WebSphere Adapters either online or by phone. Gathering information about the problem before you contact IBM Software Support can dramatically increase support responsiveness.

Before you begin

If you think your problem is defect-related, IBM Software Support provides assistance. 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, Linux®, or UNIX® operating systems), you must be enrolled in Passport Advantage®. You can enroll in one of the following ways:

Online

Go to the Passport Advantage Web page (<http://www-306.ibm.com/software/support/pa.html>), 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 (<http://techsupport.services.ibm.com/guides/contacts.html>), 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 (<http://www-03.ibm.com/servers/eserver/techsupport.html>).

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

(<http://techsupport.services.ibm.com/guides/contacts.html>), and click the name of your geographic region for phone numbers of people who provide support for your location.

About this task

The IBM Software Support Handbook contains detailed information about the service and support of your IBM products. Read the handbook at <http://techsupport.services.ibm.com/guides/handbook.html>.

To contact IBM Software Support, use the following procedure.

How to perform this task

1. Describe your problem and gather background information. When explaining a problem to a support specialist, be as specific as possible. Include all relevant background information so that the 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? Include the version of the operating system as well as related products.
 - Has the problem happened before, or is this an isolated problem?
 - What steps led to the failure?
 - Can the problem be recreated? If so, what steps led to the failure?
 - Have any changes been made to the system such as to the hardware, operating system, networking software, and so on?
 - Are you currently using a workaround for this problem? If so, be prepared to explain it when you report the problem.
 - Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
2. Determine the business impact of your problem. When you report a problem, 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 criteria described in the following table.

Table 8. Severity criteria for problem reporting

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

3. Submit your problem to IBM Software Support. You can submit your problem in the following ways:
 - **Online.** Go to the Submit and track problems page on the IBM Software Support site <http://www.ibm.com/software/support/probsub.html> 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 (<http://techsupport.services.ibm.com/guides/contacts.html>), and click the name of your geographic region.

Result

If the problem you submit is for an unreported software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail and tracks its resolution.

What to do next

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

Chapter 10. Quick start tutorials

To gain practical knowledge in setting up and deploying the adapter, complete one or more of the tutorials. Everything you need to complete each tutorial is contained in the tutorial. If you have performed the prerequisite tasks (such as installing the adapter), you can complete each tutorial in under an hour.

Introduction

Each scenario provides a complete set of instructions for configuring the adapter so that it can be used by a J2EE component to send requests to the enterprise information file system or by the enterprise information file system to send requests to a J2EE component. You use WebSphere Integration Developer (and its enterprise service discovery wizard) to configure the adapter, connect to the enterprise information file system, and retrieve information about a service or services. Enterprise service discovery creates the business objects and interface information needed to interact with the service, all of which are built, along with the adapter, into a deployable module.

The tutorials consists of three individual scenarios:

- Outbound processing of event files with data transformation
- Inbound processing of event files with data transformation
- Splitting of event files during inbound processing

Learning objectives

After completing the tutorial, you should be able to perform the following tasks:

- Create an adapter project in WebSphere Integration Developer
- Discover services and associated business objects from the enterprise information file system and make them part of the adapter project
- Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus
- Test the module to ensure that it operates correctly and to see the results of running the module

Time required

Each scenario in this tutorial should take approximately thirty minutes to finish.

Audience

These tutorials are for integration developers who design, assemble, test, and deploy business integration solutions.

Prerequisites

To complete this tutorial, the following applications must be either installed or accessible and the following folders must be created:

- WebSphere Integration Developer, version 6.0.2
- WebSphere Adapter for Flat Files
- Output directory must be created at C:\flatfiledir\outputdir

- Staging directory must be created at C:\flatfiledir\stagingdir
- Event directory must be created.

Tutorial 1: Outbound processing with data transformation

The outbound scenario demonstrates how the adapter generates output events in a specified folder with data transformation for the create, append, overwrite, delete, list, retrieve, and exists operation.

Creating the adapter project in WebSphere Integration Developer

Use WebSphere Integration Developer to create a connector project and add the RAR file to the project.

1. In WebSphere Integration Developer, switch to the J2EE perspective:
 - a. Click **Window** → **Open Perspective** → **Other**.
 - b. Click **J2EE**.
If **J2EE** is not displayed, select the **Show all** check box, click **J2EE**, and click **OK**.
 - c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.
 - d. Click **OK**.
2. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

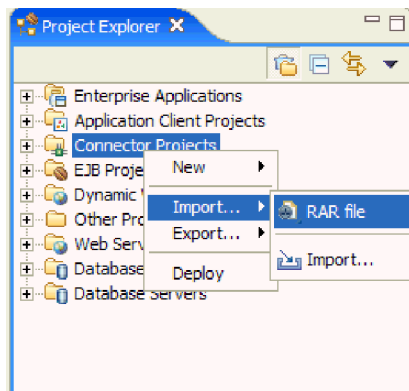


Figure 24. Importing the RAR file

3. In the Connector import window, click **Browse**, and select the RAR file location. For example, C:\IBM\ResourceAdapters\FlatFiles\adapter\flatfile\deploy.
4. In the **Connector project** field, specify a project name.
5. Clear the **Add module to an EAR project** check box.

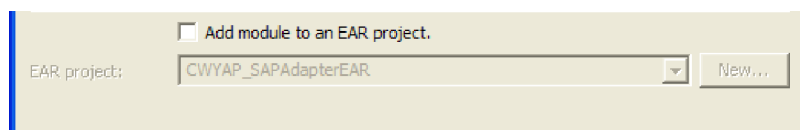


Figure 25. Clearing the Add module to an EAR project check box

6. Accept all other defaults.
7. Click **Finish**.

Result

A new J2EE connector project, named CWYFF_FlatFile, is created in your WebSphere Integration Developer workspace.

Configuring the adapter for outbound processing

Use the enterprise service discovery tool in WebSphere Integration Developer to set the connection properties for the adapter, select business objects or services that are in the enterprise information system, and generate business object definitions and related artifacts for outbound processing.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to set the connection properties necessary to enable communication between the adapter and the enterprise information file system (EIS). In this tutorial you will enable the data transformation capabilities of the adapter by specifying optional values that will convert unstructured data into a business object. For your actual implementation, you avoid enabling data transformation (creating a pass-through scenario) by retaining all of the defaults and clicking next.

1. In WebSphere Integration Developer, start the enterprise service discovery wizard:
 - a. Switch to the Business Integration Perspective by clicking **Window** → **Open** → **Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
 - b. Click **File** → **New** → **Enterprise Service Discovery** and click **Next**.

Note: If **Enterprise Service Discovery** is not displayed, select the **Show all wizards** checkbox, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

2. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for Flat Files (version 6.0.2) from the 'CWYFF_FlatFile' Connector Project** and click **Next**.

Note: If the 'CWYFF_FlatFile' Connector Project isn't listed, complete the following steps to make it appear:

- a. Click **Import Resource Adapter**.
 - b. Browse for the Connector file CWYFF_FlatFile.rar.
 - c. Select a target server type from the drop down menu.
 - d. Click **Finish**.
3. Optional: If you are asked if you want to switch to the J2EE perspective, click **Yes**.
 4. Complete the following steps in the Configure Settings for Discovery Agent window to enable data transformation.
 - a. For the **Folder name** field, browse to the path C:\Program Files\IBM\ResourceAdapters\FlatFiles\adapter\flatfile\samples. This is where the tutorial XSD schemas for the business objects are stored in the default installation. For this tutorial, you will use the sample Customer.xsd
 - b. Leave the **Character Set** field empty.

- c. From the **Content Type** list, select **text/xml** as the format to be used for all business objects. This is a one time setting and is used to bundle a content type with a corresponding data binding.
- d. When you set the **Content Type** value, the **DataBindingType** field is populated with XMLBOSerializerDataBinding automatically. This is the name of the data binding corresponding to the content type.

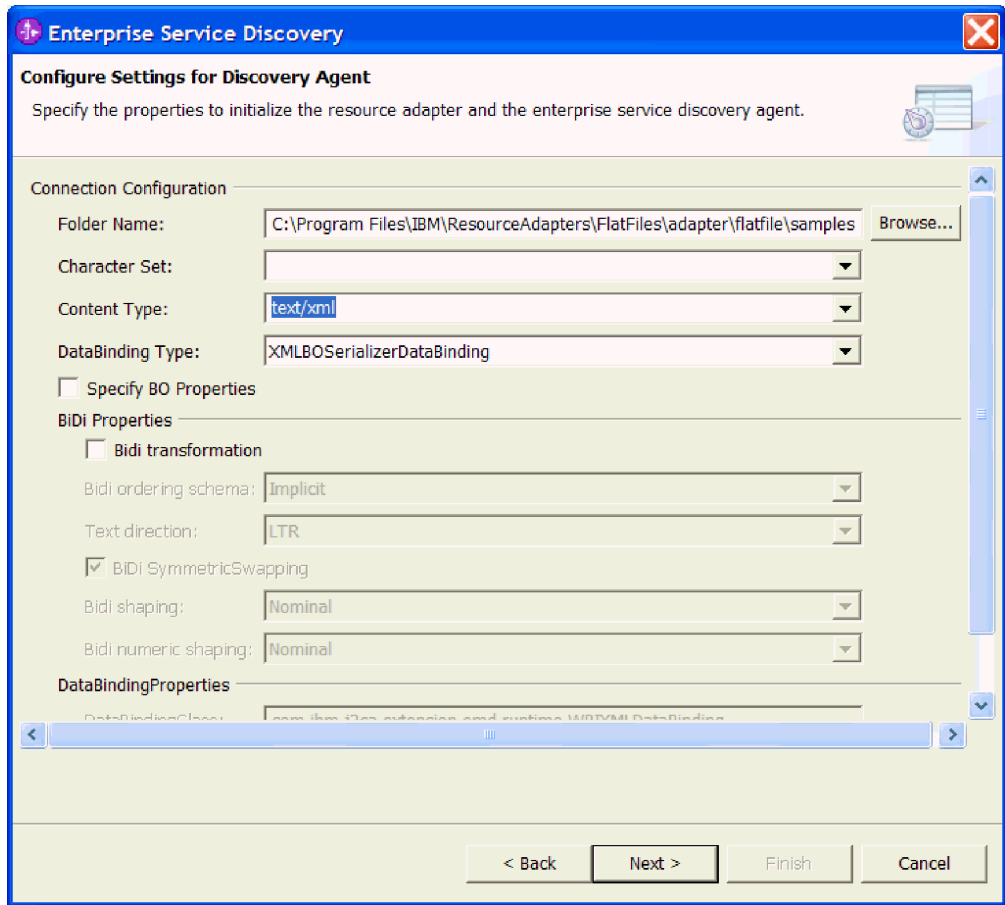


Figure 26. The Configure Settings for Discovery Agent window

- e. Click **Next**.
5. Set the logging level so that you can see any errors that might arise during configuration.
 - a. Click **Show Advanced**. The button changes to **Hide Advanced**.
 - b. In the **Log file output location** field, accept the default path of the log file.
 - c. From the **Logging Level** list, select **FINEST**.
6. Click **Next**.

Result

The connection properties that the enterprise service discovery wizard uses to communicate with the enterprise information system are set. The wizard advances to the Find and Discover Enterprise Services screen.

Selecting the business objects and services to be used with the adapter

Use the enterprise service discovery wizard to browse for business objects and other metadata information on the enterprise information system. Then, select artifacts used for outbound processing.

1. In the Find and Discover Enterprise Services window, click **Execute Query** to display the business objects for the adapter.
2. In the "Objects discovered by query" pane, highlight the **Customer** business object, then click **Add to import list** to move the business object to the "Objects to be imported" pane.

Note: To remove objects from the "Objects to be imported" pane, highlight the object that you want to remove and click **Remove**.

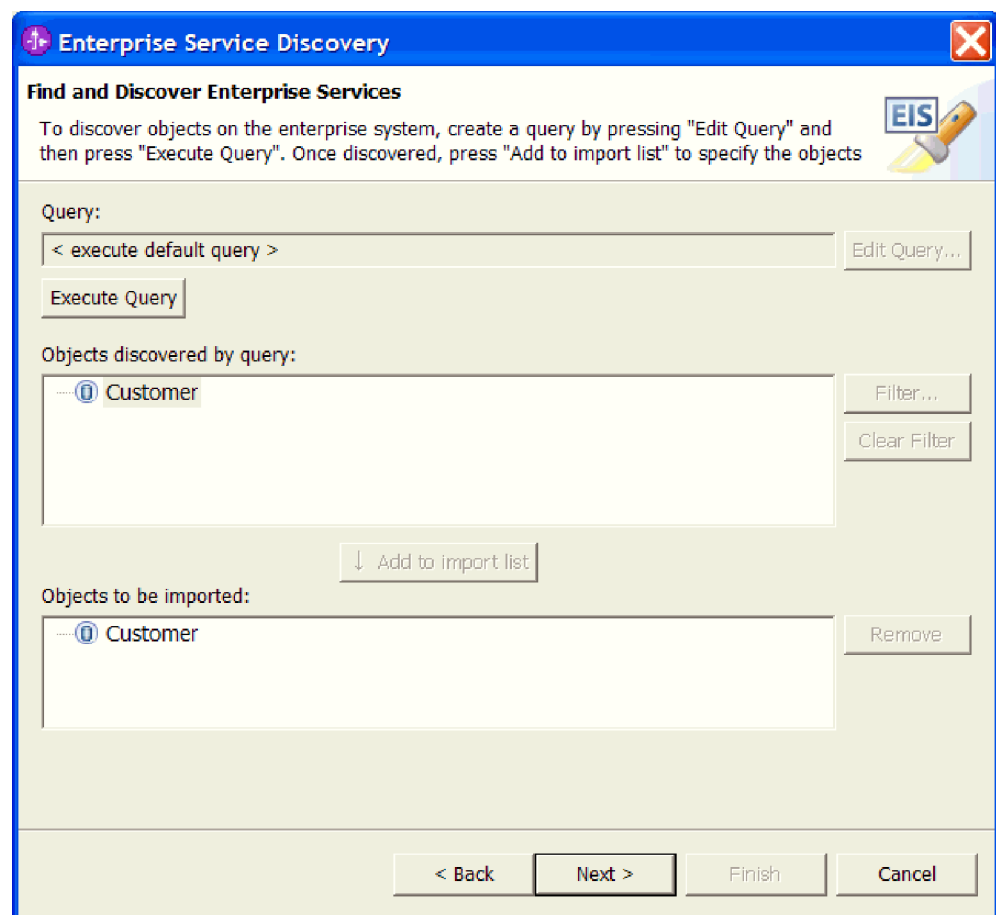


Figure 27. The Find and Discover Enterprise Services window

3. Click **Next**.

Result

You have selected the business objects that will be used in the tutorial outbound connector project.

Configuring the selected objects

Once you have added the business object to the module, configure it for outbound operations.

1. In the Configure Objects window of the enterprise service discovery wizard, select **Outbound** from the **Service Type** list. The default base namespace for the business object schema to be generated is displayed. This value can be changed.
2. Type the location of the business object in the **BO Location** field. This creates the specified directory name in your connector project.
3. Click **Next**. All of the listed operations are selected by default. You can change the list by clicking the **Add** or **Remove** buttons.

The objects are now configured for outbound communication.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as an SCA module.

1. To create a new business integration module, use the following procedure.
 - a. In the Generate Artifacts window, next to the **Module name** field, click **New**.
 - b. In the Integration Project window, select the default setting, **Create a module project**, and click **Next**.
 - c. In the New Module window, type "FlatFileOutboundModule" in the **Module Name** field.
 - d. Under Module Location, select the **Use Default** check box.
 - e. Click **Finish**.
2. In the Generate Artifacts window, select **Use discovered connection properties**.

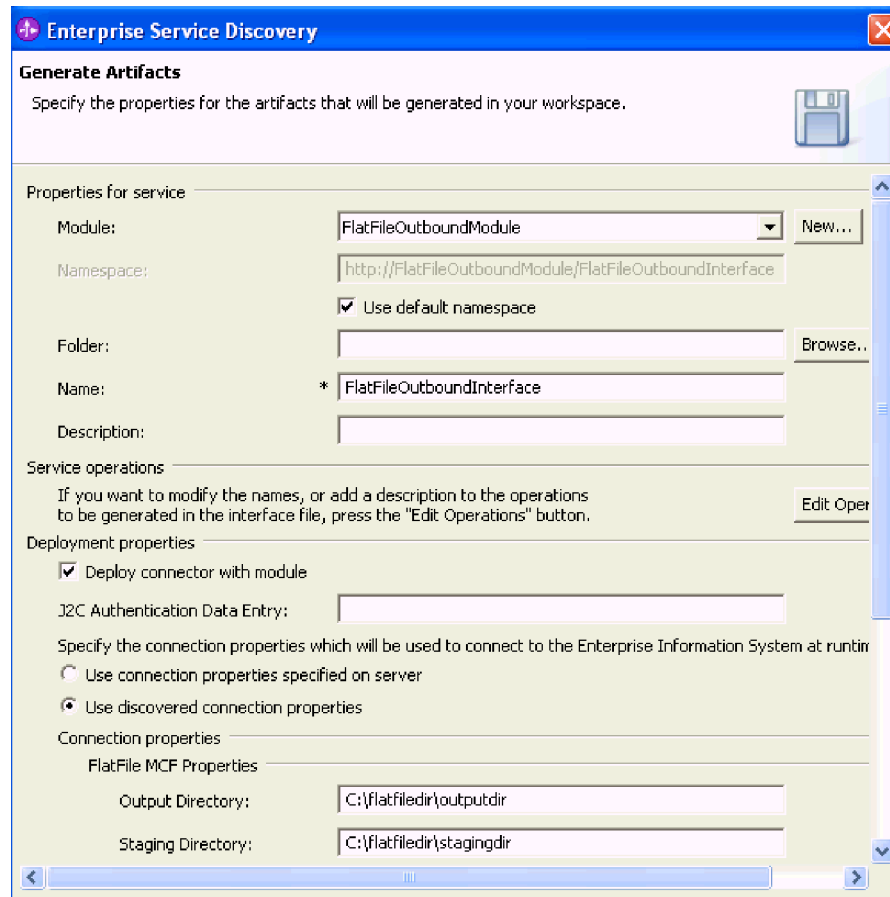


Figure 28. Generate Artifacts window

3. In the Generate Artifacts window, type the values for the managed connection factory properties. These properties are used by the adapter to create outbound connections to the enterprise information system file repository.
 - a. In the **Output Directory** field, type the path C:\flatfiledir\outputdir. This directory must be created on your file system for this value to work. This value tells the adapter where it will write output files during outbound processing.
 - b. In the **Staging Directory** field, type the path C:\flatfiledir\stagingdir. This directory must be created on your file system for this value to work. This value tells the adapter where it will create, append, and overwrite operations during outbound processing.

Note: This step is not required in an actual implementation, but included in this tutorial for educational purposes.

4. Specify the following remaining values as follows:

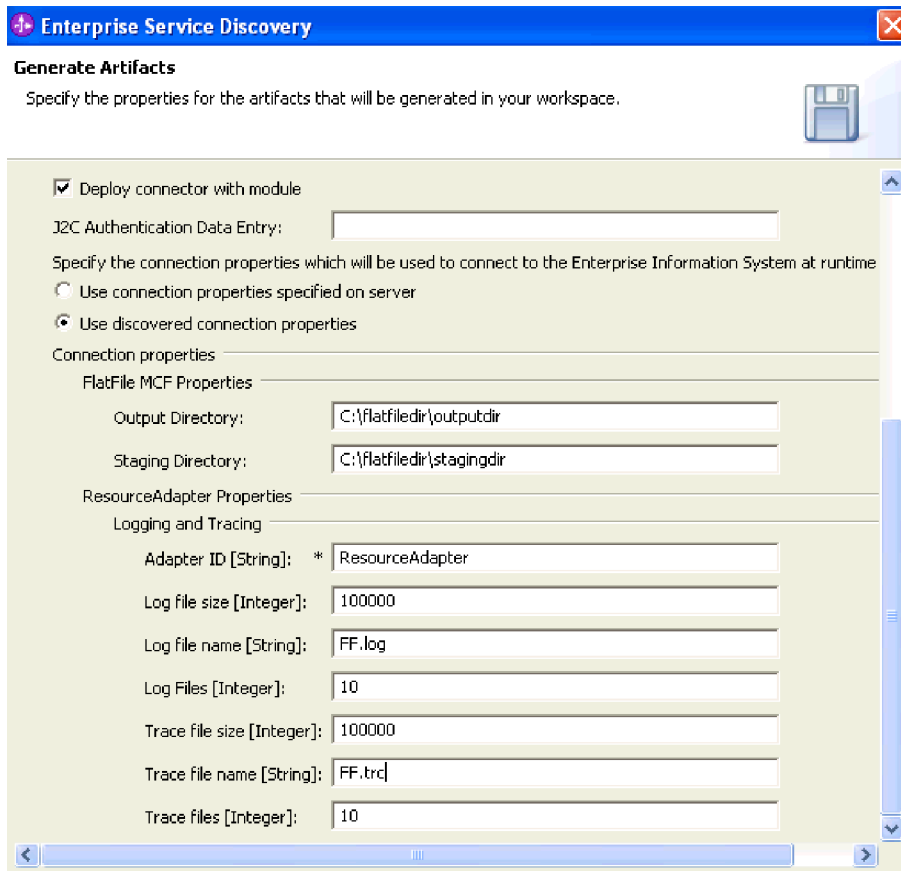


Figure 29. Lower half of the Generate Artifacts window

- a. In the **Adapter ID** field, retain the default.
 - b. In the **Log file size** field, type 100000. This indicates the size for each log file in kilobytes. If no value is specified, the file will have no maximum size.
 - c. In the **Log file name** field, type c:\logs as the full path of the log file.
 - d. In the **Log files** field, type 10 for the maximum number of log files to use after the log file has reached its maximum size. When a log file reaches its maximum size it will start using another log file. If no value is specified it will be set to 1.
 - e. In the **Trace file size** field, type 100000 for the size for each trace file in kilobytes. If no value is specified, the file will have no maximum size.
 - f. In the **Trace file name** field, type c:\trace as the full path of the trace file.
 - g. In the **Trace files** field, enter 10 for the maximum number of trace files to use after the log file has reached its maximum size. When a trace file reaches its maximum size it will start using another log file. If a value is not specified it will be set to 1.
5. Click **Finish**.

Result

The FlatFileOutboundInterface.wsdl, FlatFileOutboundInterface.import artifacts, FlatFileBG, FlatFile, UnstructuredContent, CustomerWrapperBG, CustomerWrapper, and Customer business objects are generated. The application business objects specified by the user are updated with application-specific information for data transformation and saved in the business object location.

Testing outbound with data transformation for the create operation

Test the create operation with the outbound module created in Tutorial 1 using the WebSphere Integration Developer test module.

1. If WebSphere Process Server is not started, use the following procedure to start WebSphere Process Server in WebSphere Integration Developer:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window** → **Show View** → **Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
2. Switch to the Business Integration perspective.
3. In WebSphere Integration Developer, right click **FlatFileOutboundModule** and select **Test** → **Test Module**.
4. Expand **Detailed Properties**.
5. In the Events window, select **createCustomer** from the **Operation** list and provide the following **Initial request parameters**:

Name	Value
customerName	Any name
Address	Any address
City	Any City
State	Any State
fileName	filecreate.txt
directoryPath	C:\flatfiledir\outputdir
fileContentEncoding	UTF-8
includeEndBODelimiter	<null>
Staging directory	<null>

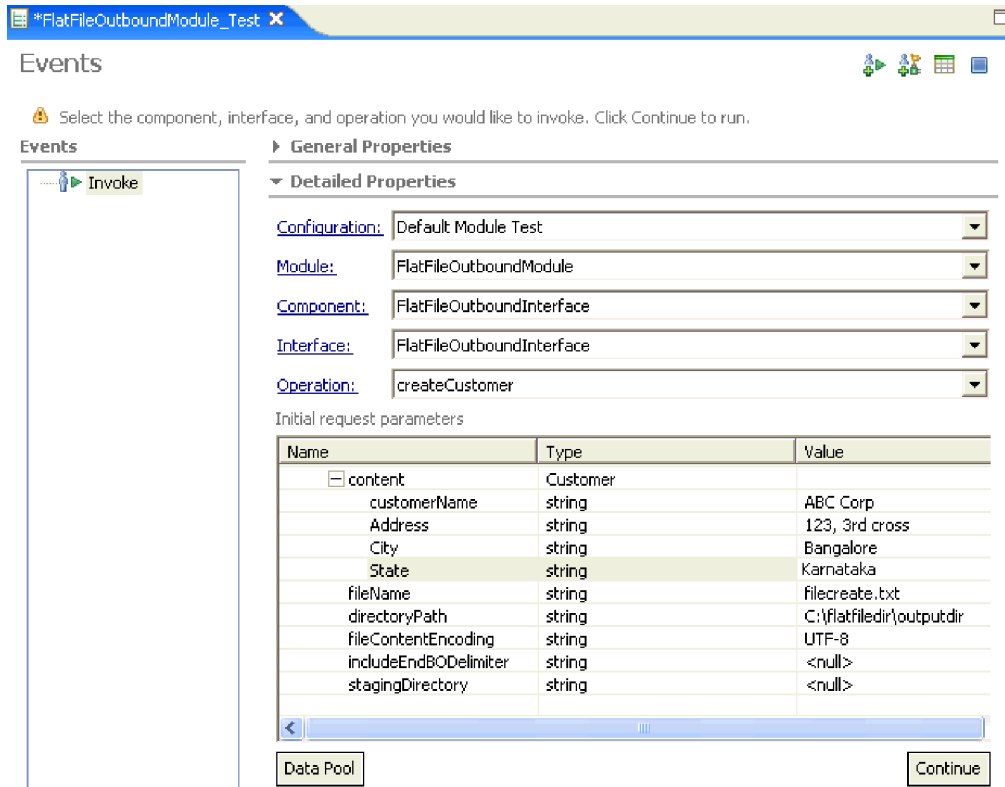


Figure 30. Events page with the createCustomer operation selected

6. Click **Continue**.
7. Optional: If asked to verify your server deployment, select your server from the list and click **Finish**.

Result

Verify that the file has been created in the C:\flatfiledir\outputdir directory with the specified contents in text/xml format.

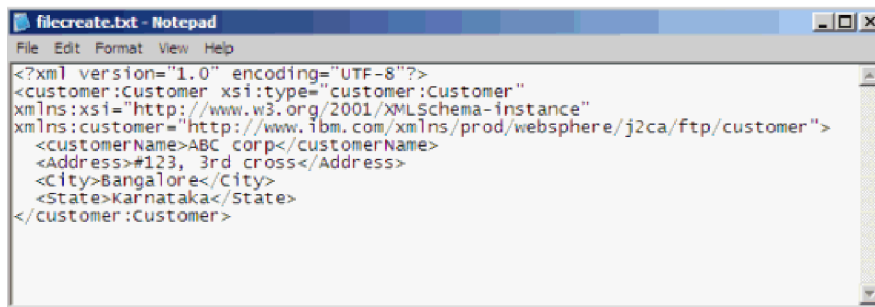


Figure 31. Contents of the filecreate.txt file

Testing outbound with data transformation for the exists operation

Test the existsCustomer operation with the outbound module created in Tutorial 1 using the WebSphere Integration Developer test module.

1. If WebSphere Process Server is not started, use the following procedure to start WebSphere Process Server in WebSphere Integration Developer:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window** → **Show View** → **Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
2. Switch to the Business Integration perspective.
3. In WebSphere Integration Developer, right click **FlatFileOutboundModule** and select **Test** → **Test Module**.
4. Expand **Detailed Properties**.
5. In the Events window, select **existsCustomer** from the **Operation** list and provide the following **Initial request parameters**:

Name	Value
customerName	Leave empty
Address	Leave empty
City	Leave empty
State	Leave empty
fileName	filecreate.txt
directoryPath	C:\flatfiledir\outputdir
fileContentEncoding	UTF-8
includeEndBODelimiter	<null>
Staging directory	<null>

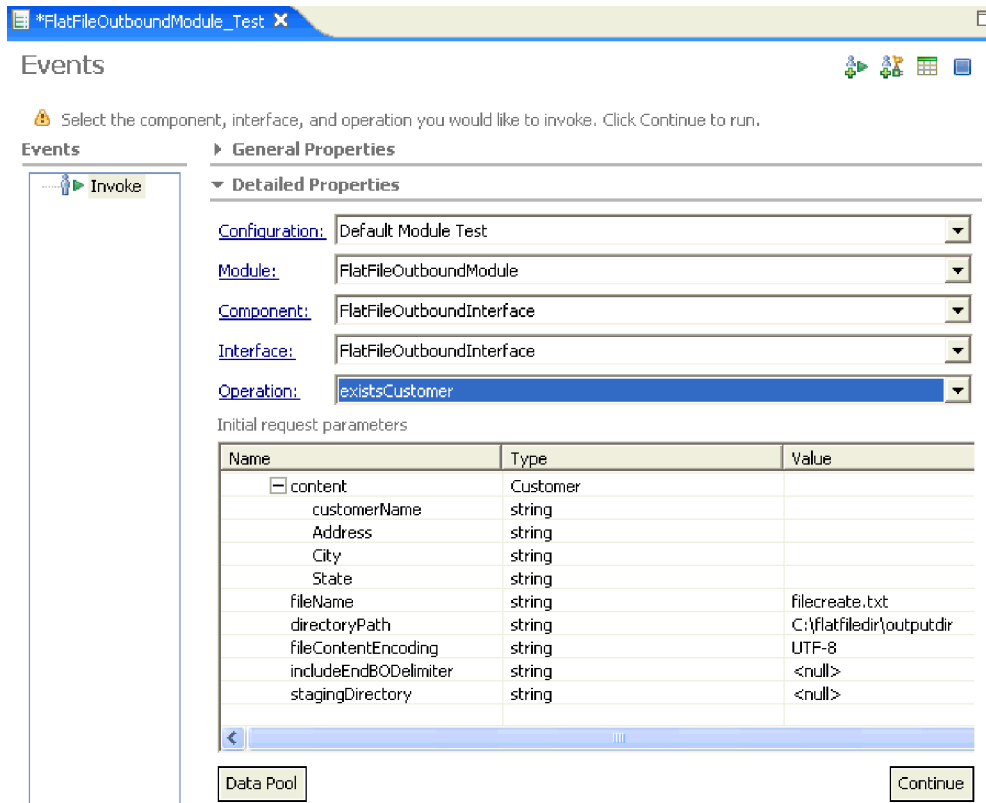


Figure 32. Events page with the existsCustomer operation selected

6. Click **Continue**.

Result

You will see the following screen with the ExistsResponseBG returned and the value of doesFileExist is true, indicating that file exists in the specified directory.

► General Properties
▼ Detailed Properties
 Module: [FFOModule](#)
 Component: [FlatFileOutboundInterface](#)
 Interface: [FlatFileOutboundInterface](#)
 Operation: [existsCustomer](#)
 Return parameters:

Name	Type	Value
<input type="checkbox"/> existsCustomerOutput	ExistsResponseBG	
verb	VerbType	<null>
<input type="checkbox"/> ExistsResponse	ExistsResponse	
doesFileExist	Boolean	true

Figure 33. Window containing the existsCustomer results

Repeat the above test for a file name that does not exist and verify that the returned business object’s value for doesFileExist is false.

Testing outbound with data transformation for the list operation

Test the listCustomer operation with the outbound module created in Tutorial 1 using the WebSphere Integration Developer test module.

1. If WebSphere Process Server is not started, use the following procedure to start WebSphere Process Server in WebSphere Integration Developer:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window → Show View → Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
2. Switch to the Business Integration perspective.
3. In WebSphere Integration Developer, right click **FlatFileOutboundModule** and select **Test → Test Module**.
4. Expand **Detailed Properties**.
5. In the Events window, select **listCustomer** from the **Operation** list and provide the following **Initial request parameters**:

Name	Value
customerName	Leave empty
Address	Leave empty
City	Leave empty
State	Leave empty

Name	Value
fileName	null
directoryPath	C:\FlatFile
fileContentEncoding	<null>
includeEndBODelimiter	<null>
Staging directory	<null>

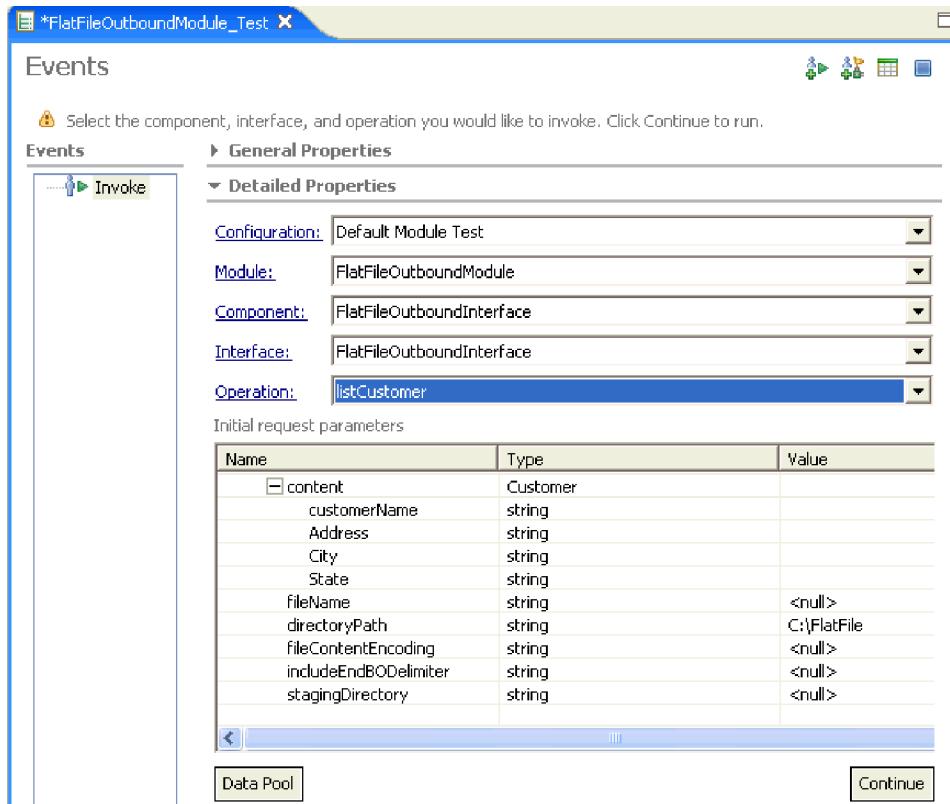


Figure 34. Events page with the listCustomer operation selected

6. Click **Continue**.

Result

The ListResponseBG business graph is returned as the response and the listOfFileNames value contains the list of files and directories present in the given directory.

General Properties		
Detailed Properties		
Module:	FFModule	
Component:	FlatFileOutboundInterface	
Interface:	FlatFileOutboundInterface	
Operation:	listCustomer	
Return parameters:		
Name	Type	Value
listCustomerOutput	ListResponseBG	
verb	VerbType	<null>
ListResponse	ListResponse	
listOfFileNames	String []	
listOfFileNames[0]	String	archivedir
listOfFileNames[1]	String	CWYFF_FlatFile.rar
listOfFileNames[2]	String	CWYFF_FlatFile.rar.1
listOfFileNames[3]	String	eventdir
listOfFileNames[4]	String	FFModule.jar
listOfFileNames[5]	String	FFModuleApp.ear
listOfFileNames[6]	String	FFModuleApp.ear.1
listOfFileNames[7]	String	FFInboundApp.ear
listOfFileNames[8]	String	outputdir

Figure 35. Window containing the listCustomer results

Testing outbound with data transformation for the retrieve operation

Test the retrieveCustomer operation with the outbound module created in Tutorial 1 using the WebSphere Integration Developer test module.

1. If WebSphere Process Server is not started, use the following procedure to start WebSphere Process Server in WebSphere Integration Developer:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window** → **Show View** → **Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
2. Switch to the Business Integration perspective.
3. In WebSphere Integration Developer, right click **FlatFileOutboundModule** and select **Test** → **Test Module**.
4. Expand **Detailed Properties**.
5. In the Events window, select **retrieveCustomer** from the **Operation** list and provide the following **Initial request parameters**:

Name	Value
customerName	Leave empty
Address	Leave empty
City	Leave empty
State	Leave empty

Name	Value
fileName	filecreate.txt
directoryPath	C:\flatfiledir\outputdir
fileContentEncoding	<null>
includeEndBODelimiter	<null>
Staging directory	<null>

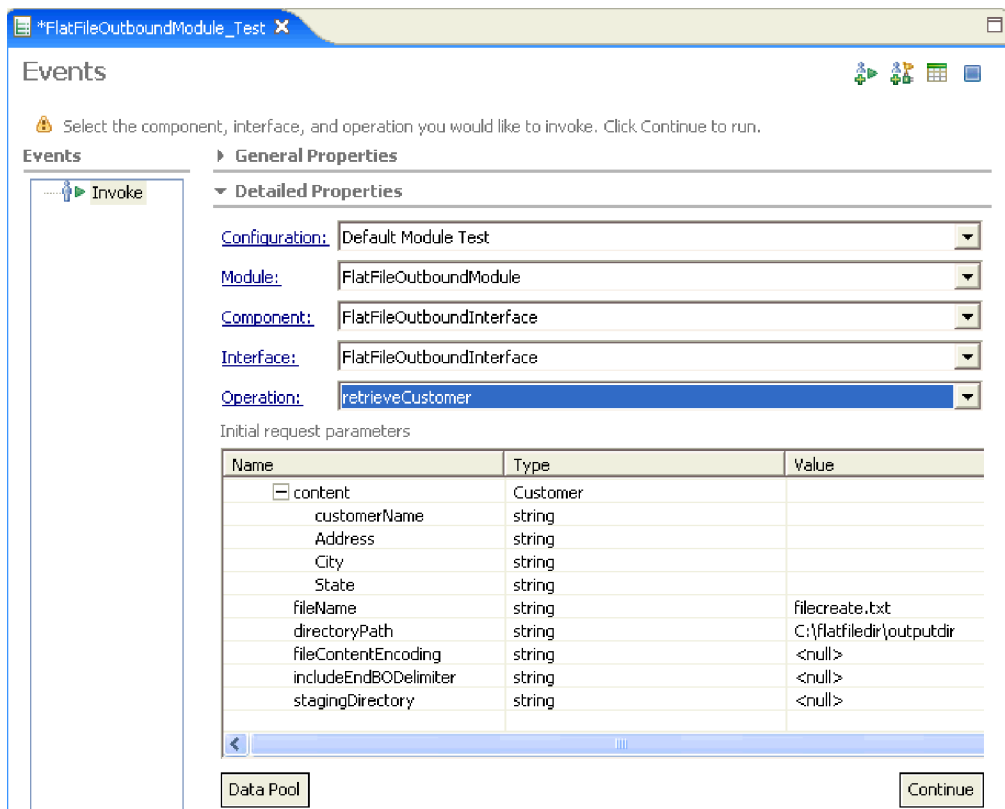


Figure 36. Events page with the retrieveCustomer operation selected

6. Click **Continue**.

Result

The RetrieveResponseWrapperBG business graph is returned and the fileContent property contains the file content.

General Properties		
Detailed Properties		
Module:	FFQModule	
Component:	FlatFileOutboundInterface	
Interface:	FlatFileOutboundInterface	
Operation:	retrieveCustomer	
Return parameters:		
Name	Type	Value
[-] retrieveCustomerOutput	RetrieveResponseWrapperBG	
verb	VerbType	<null>
[-] RetrieveResponseWra...	RetrieveResponseWrapper	
[-] Content	Object []	
[-] Content[0]	FileContent	
fileName	String	filecreate.txt
[-] fileContent	UnstructuredContent	
ContentType	String	<null>
ObjectType	String	<null>
AsText	String	<null>
AsBinary	Bytes	[B@5013f67b

Figure 37. Window containing the retrieveCustomer results

Testing outbound with data transformation for the append operation

Test the appendCustomer operation with the outbound module created in Tutorial 1 using the WebSphere Integration Developer test module.

1. If WebSphere Process Server is not started, use the following procedure to start WebSphere Process Server in WebSphere Integration Developer:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window** → **Show View** → **Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
2. Switch to the Business Integration perspective.
3. In WebSphere Integration Developer, right click **FlatFileOutboundModule** and select **Test** → **Test Module**.
4. Expand **Detailed Properties**.
5. In the Events window, select **appendCustomer** from the **Operation** list and provide the following **Initial request parameters**:

Name	Value
customerName	Any name
Address	Any address
City	Any City
State	Any State
fileName	filecreate.txt

Name	Value
directoryPath	C:\flatfiledir\outputdir
fileContentEncoding	<null>
includeEndBODelimiter	<null>
Staging directory	<null>

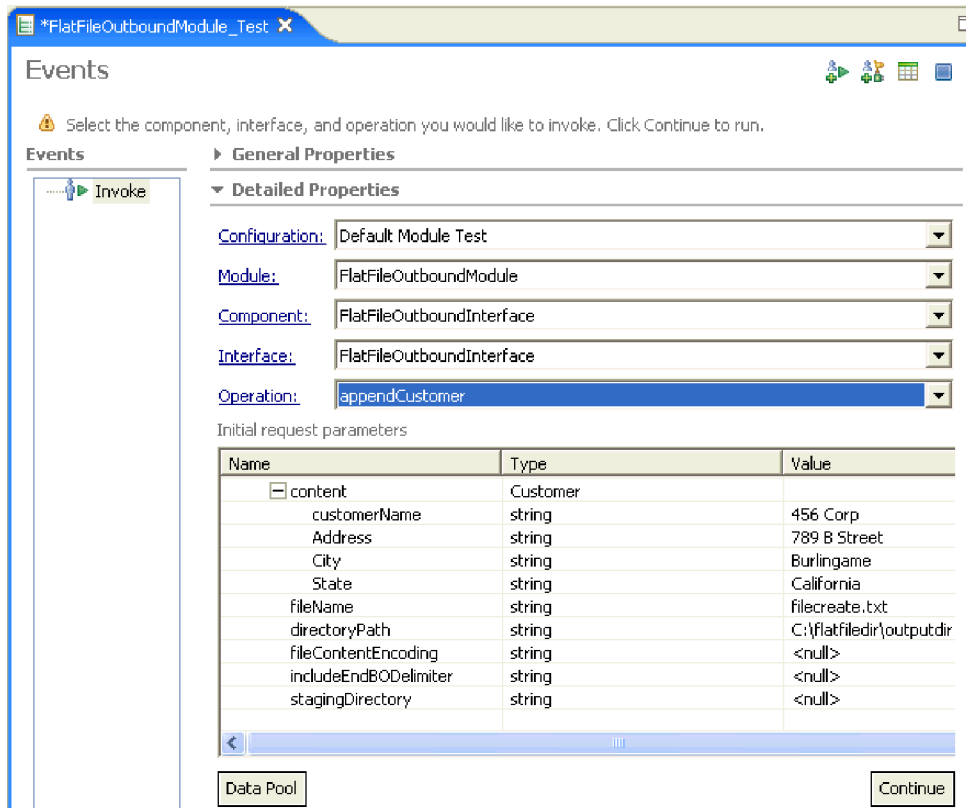


Figure 38. Events page with the appendCustomer operation selected

6. Click **Continue**.

Result

Specified content is appended to the specified file.

Testing outbound with data transformation for the overwrite operation

Test the overwriteCustomer operation with the outbound module created in Tutorial 1 using the WebSphere Integration Developer test module.

1. If WebSphere Process Server is not started, use the following procedure to start WebSphere Process Server in WebSphere Integration Developer:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window** → **Show View** → **Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
2. Switch to the Business Integration perspective.

3. In WebSphere Integration Developer, right click **FlatFileOutboundModule** and select **Test** → **Test Module**.
4. Expand **Detailed Properties**.
5. In the Events window, select **overwriteCustomer** from the **Operation** list and provide the following **Initial request parameters**:

Name	Value
customerName	Any name
Address	Any address
City	Any City
State	Any State
fileName	filecreate.txt
directoryPath	C:\flatfiledir\outputdir
fileContentEncoding	<null>
includeEndBODelimiter	<null>
Staging directory	<null>

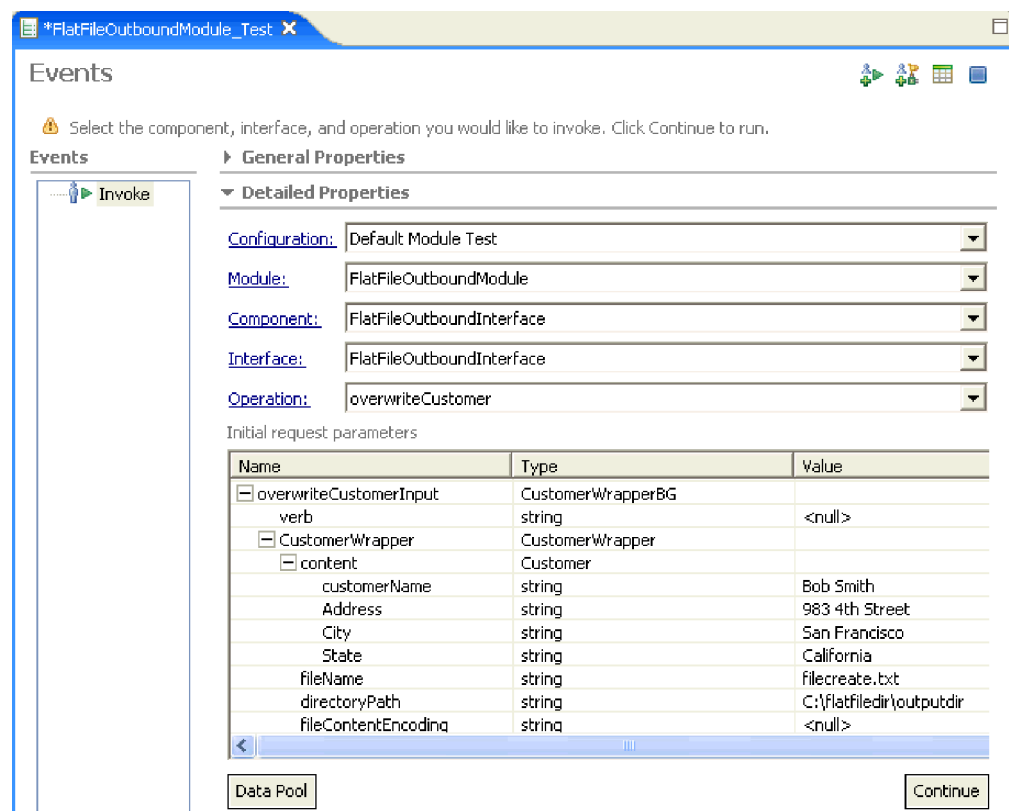


Figure 39. Events page with the *overwriteCustomer* operation selected

6. Click **Continue**.

Result

The existing content is overwritten with the specified content.

Testing outbound with data transformation for the delete operation

Test the deleteCustomer operation with the outbound module created in Tutorial 1 using the WebSphere Integration Developer test module.

1. If WebSphere Process Server is not started, use the following procedure to start WebSphere Process Server in WebSphere Integration Developer:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window** → **Show View** → **Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
2. Switch to the Business Integration perspective.
3. In WebSphere Integration Developer, right click **FlatFileOutboundModule** and select **Test** → **Test Module**.
4. Expand **Detailed Properties**.
5. In the Events window, select **deleteCustomer** from the **Operation** list and provide the following **Initial request parameters**:

Name	Value
customerName	Leave empty
Address	Leave empty
City	Leave empty
State	Leave empty
fileName	filecreate.txt
directoryPath	C:\flatfiledir\outputdir
fileContentEncoding	<null>
includeEndBODelimiter	<null>
Staging directory	<null>

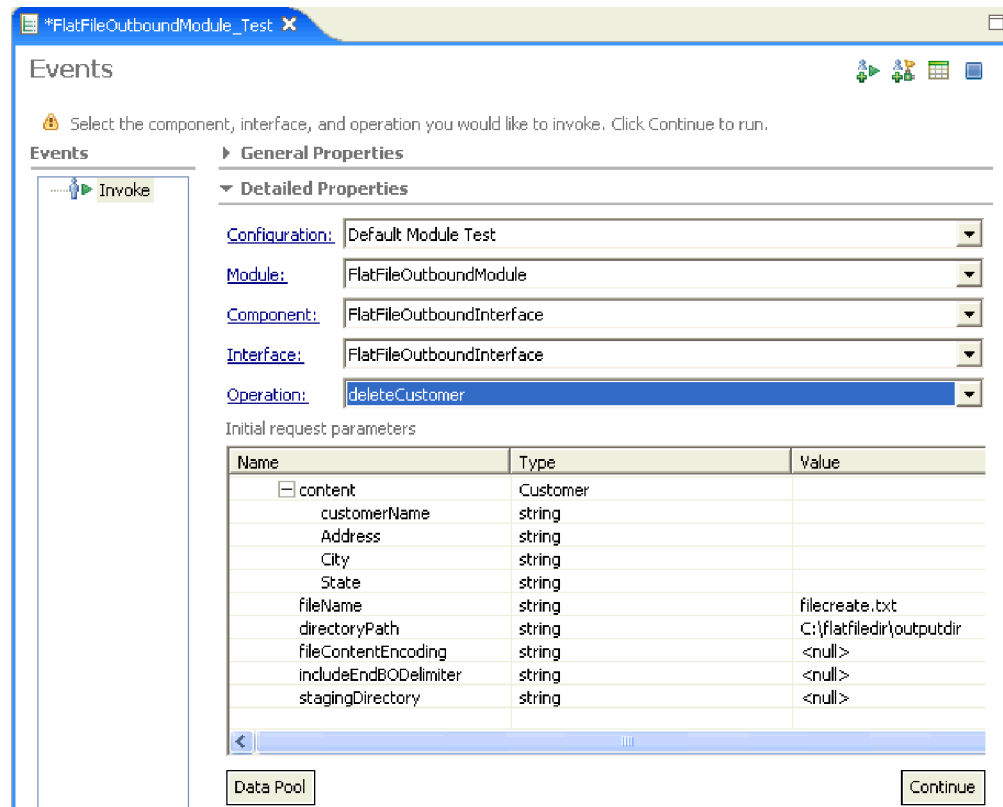


Figure 40. Events page with the deleteCustomer operation selected

6. Click **Continue**.

Result

The specified file is permanently deleted from the file system. Verify that the filecreate.txt file has been deleted from the C:\flatfiledir\outputdir directory.

Tutorial 2: Inbound processing with data transformation

The inbound scenario demonstrates how event files stored in a specified folder on an enterprise information file system are picked up by the adapter and how those event files are processed.

Creating the adapter project in WebSphere Integration Developer

Use WebSphere Integration Developer to create a connector project and add the resource adapter archive (RAR) file to the project. Importing the RAR file automatically creates a new J2EE connector project for the adapter in your workspace in WebSphere Integration Developer.

About this task

Note: If you have already created the adapter project, you do not need to do it again. You can prepare your workspace by deleting any files created during a previous tutorial.

How to perform this task

1. In WebSphere Integration Developer, switch to the J2EE perspective:
 - a. Click **Window** → **Open Perspective** → **Other**.
 - b. Click **J2EE**.
If **J2EE** is not displayed, select the **Show all** check box, click **J2EE**, and click **OK**.
 - c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.
 - d. Click **OK**.
2. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

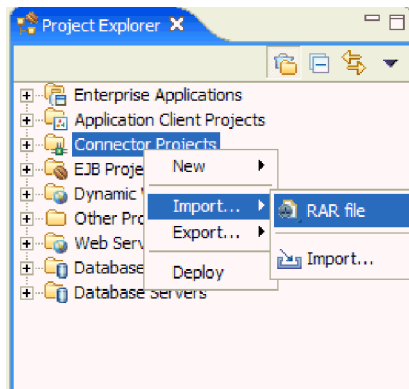


Figure 41. Importing the RAR file

3. Find the RAR file on your local file system by clicking **Browse** and navigating to the directory where the Adapter for Flat Files was installed. If you chose the default path when installing the adapter, use the following directory path:
C:\Program Files\IBM\ResourceAdapters\FlatFile\adapter\FlatFile\deploy.
4. Accept the default setting (**CWYFF_FlatFile**) for **Connector project**.

The connector project has the same name as the RAR file.

Note: If a project named CWYFF_FlatFile already exists in this workspace, the name in the Connector project field has a number appended to it (for example, CWYFF_FlatFile1).

5. Accept the default value in the **Target server** field.
The default value is the test environment for WebSphere Process Server, which is installed as part of WebSphere Integration Developer.
6. Clear the **Add module to an EAR project** check box.

Note: The EAR project field becomes unavailable after you remove the check mark.

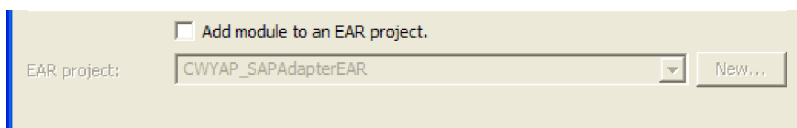


Figure 42. Clearing the Add module to an EAR project check box

7. Click **Finish**.

Result

A new J2EE connector project, named CWYFF_FlatFile, appears in the project explorer and is created in your WebSphere Integration Developer workspace.

Create the event database

This scenario relies on having a database called FFDB where the adapter will store events. After you create this database using the BAT file supplied in the samples folder, the adapter will create the event table automatically.

About this task

Note: The server must be stopped when creating the event database. You will be prompted to start the server in the next section.

How to perform this task

1. In the folder where you saved the sample files, browse for <WPS_installation_directory>\cloudscape\bin\embedded\cview.bat.
2. Run the cview.bat file. This opens the Cloudscape™ graphical user interface.
3. In the Cview window, select **File** → **New** → **Database**.
4. In the **New Database** name field, type FFDB.
5. Select **OK** and to close all windows.

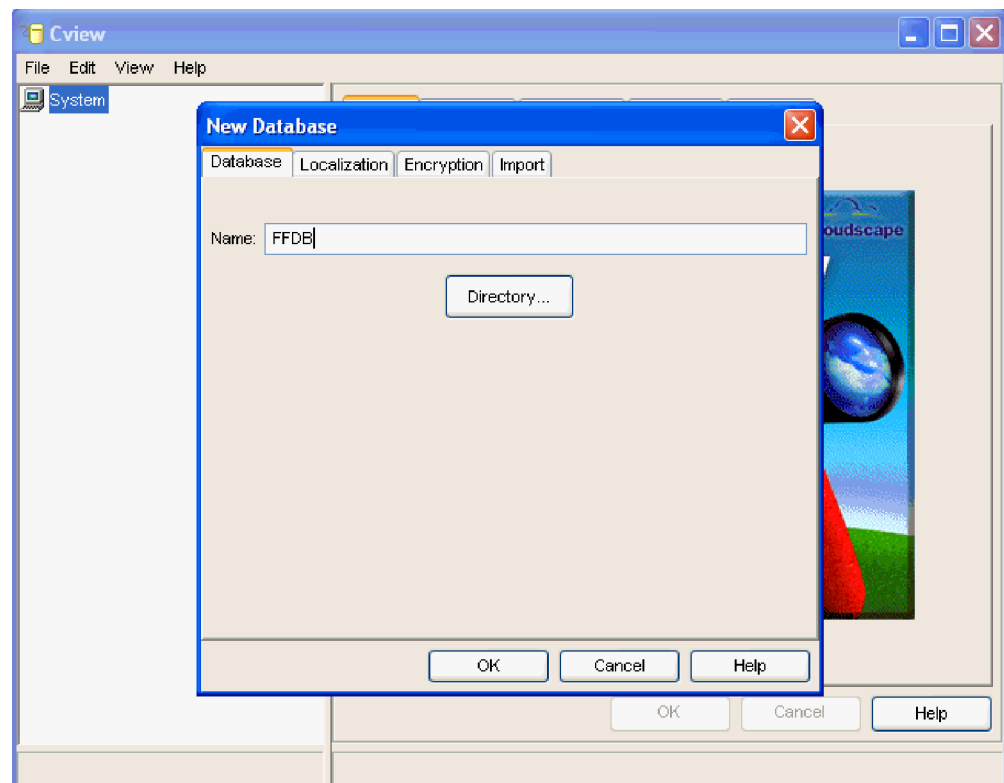


Figure 43. New database creation window

Creating and configuring the data source

Once you have created the FFDB database, you must configure WebSphere Integration developer to recognize it as the data source.

1. Open WebSphere Integration Developer. The Business Integration perspective will open by default.
2. On the bottom right corner of the Business Integration perspective, click on the **Server** tab to bring it to the front.
3. Right-click your server instance and select **Start**.
4. Confirm that the server is active and ready for transaction requests. When it is ready, the Console window will say Server <server profile> is open for business.
5. Open the test server's administrative console window by right-clicking the server profile on the **Server** tab and selecting **Run Administrative Console**. The port number for the administrative console may differ based on your setup. If your port number is different from the default, use your specific port number.
6. In the **user ID** field, enter the default administrative user ID admin and click **Log in**.
7. Create a JDBC data source in WebSphere Process Server. This tells WebSphere Process Server that the data is from FFDB, the database that you created.
 - a. In the left pane of the administrative console, select **Resources** → **JDBC Providers**. Confirm that **Node: widNode** is selected in the JDBC providers pane.
 - b. Select the check box for **Cloudscape JDBC Provider (XA)** and click **New**.

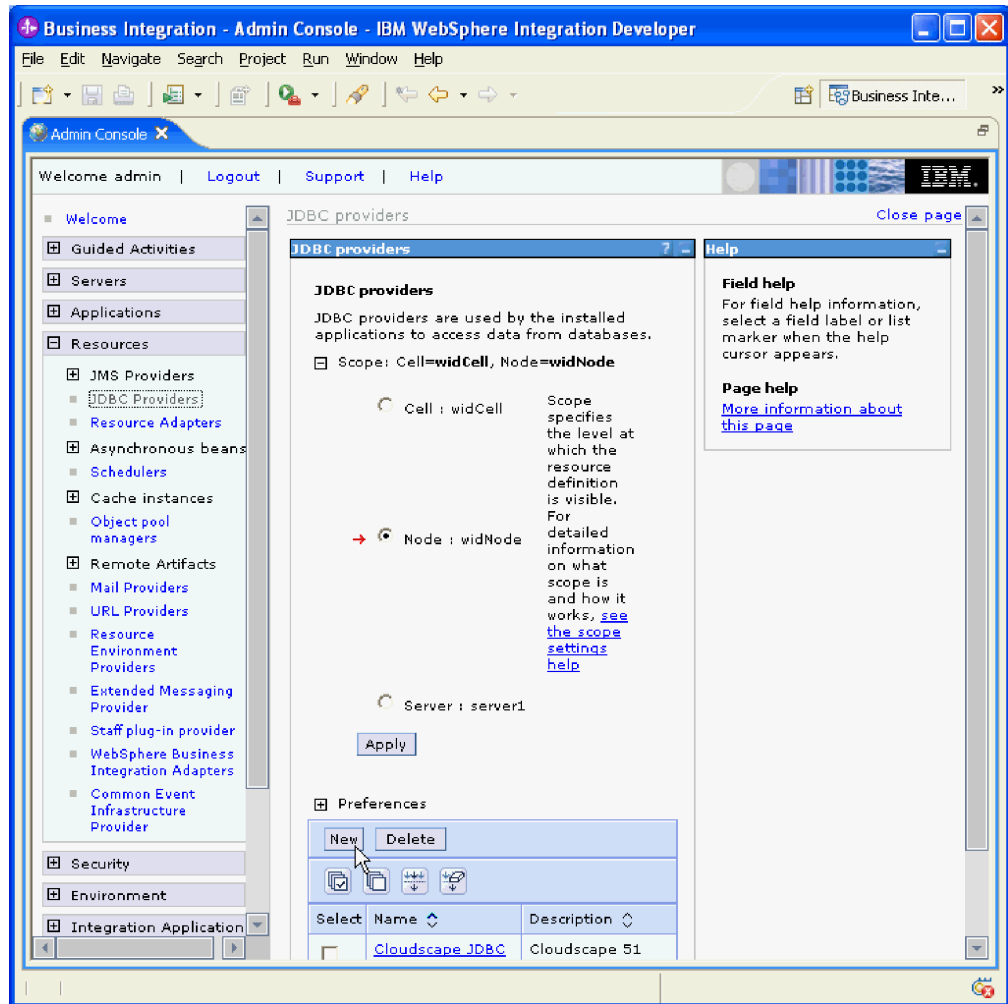


Figure 44. New JDBC providers pane

- c. In the Configuration window, select **Cloudscape** as the database type, **Cloudscape JDBC Provider** as the provider type, and **XA data source** as the implementation type. Click **Next**.

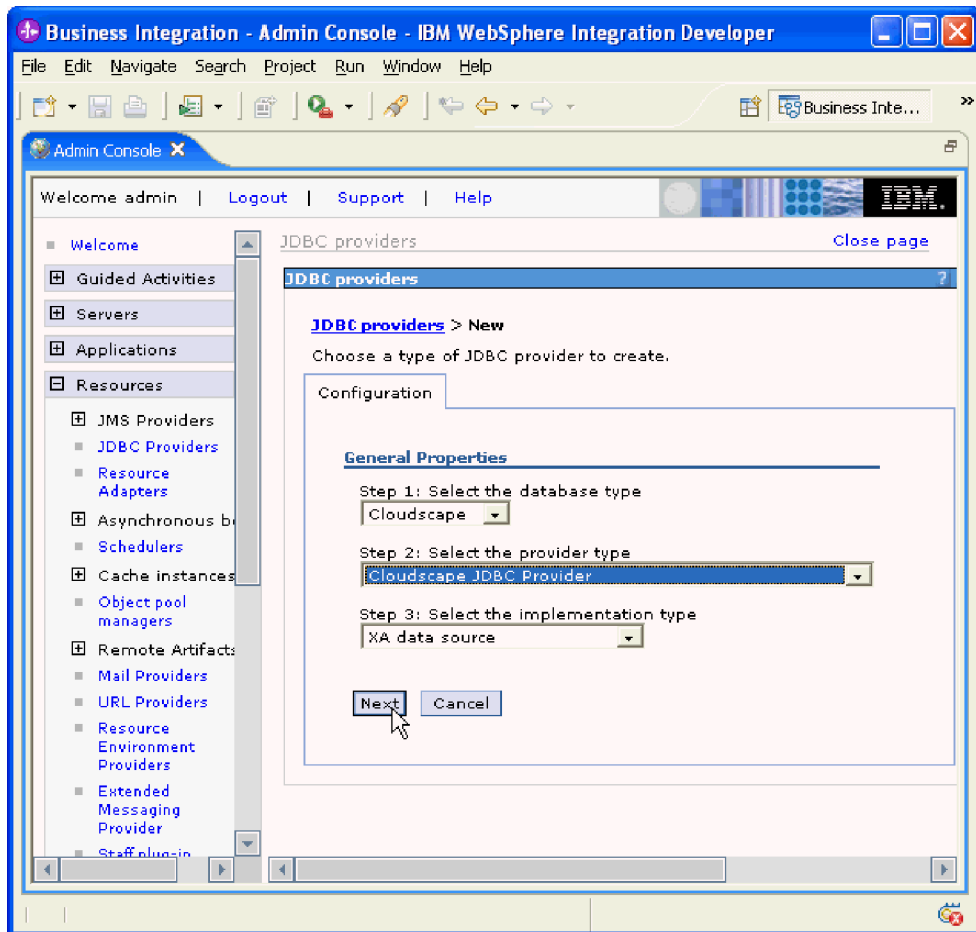


Figure 45. JDBC providers configuration pane

- d. Click **OK** in the General Properties window.
- e. Click **Cloudscape JDBC Provider (XA)**, then click **Data Sources** from the right pane of the window.

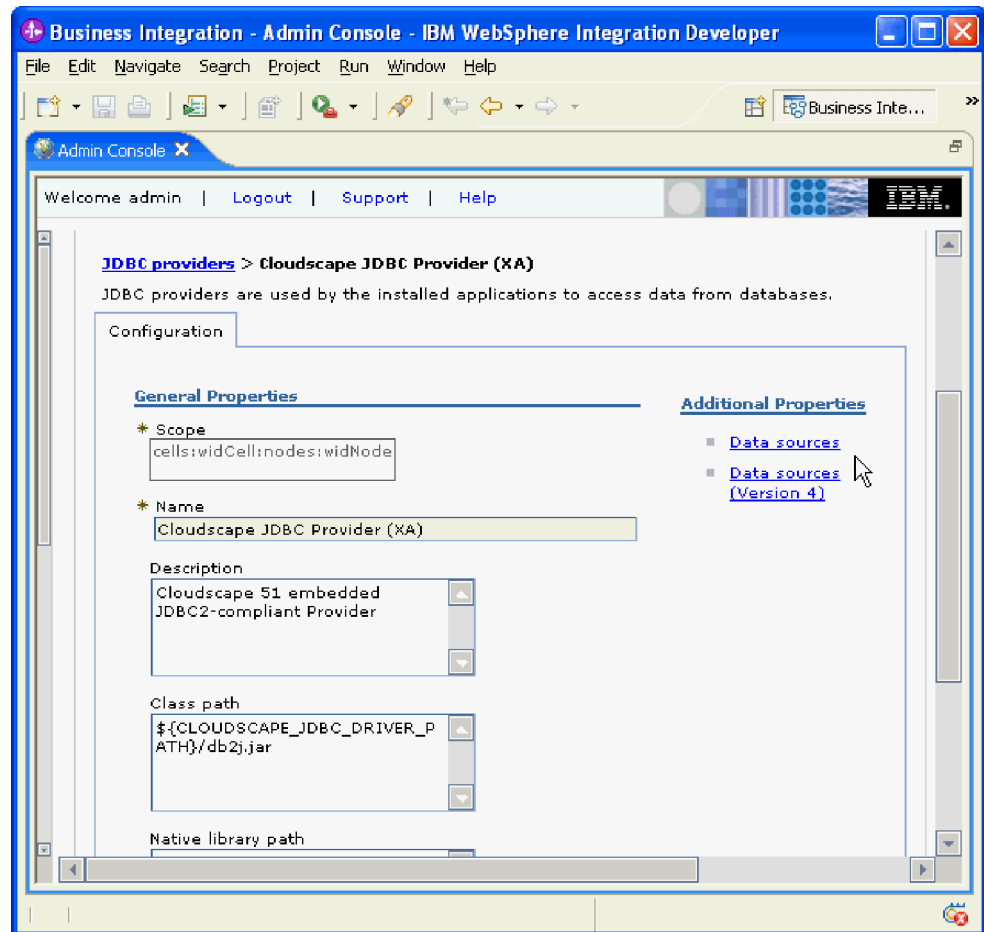


Figure 46. Data sources selection window

- f. Select **New** and then type DB2JNDI in the **JNDI name** field.
- g. Scroll down to the **Database name** field and type FFDB.
- h. Click **OK**.
8. Click **Save** to apply the changes to the master configuration.
9. Select the data source you just created and test the connection.
 - a. Select the check box next to **Cloudscape JDBC Driver XA DataSource**.
 - b. Click **Test connection**.

Result

If the connection is successful, the messages pane at the top of the window displays the following message: Test connection for data source Cloudscape JDBC Driver XA DataSource on server server 1 at node <node name> was successful.

Note: The test connection must be successful to run the rest of the tutorial.

Configuring the adapter for inbound processing

Use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties, select business objects or services that are in the enterprise information system, and generate business object definitions and related artifacts for inbound processing.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to set the connection properties necessary to enable communication with the enterprise information file system. Once communication is established between the two entities, the enterprise service discovery wizard can obtain the necessary metadata from the enterprise information file system.

1. Open WebSphere Integration Developer.
2. Optional: If WebSphere Integration Developer doesn't open in the Business Integration Perspective by default, switch by clicking **Window** → **Open** → **Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
3. Click **File** → **New** → **Enterprise Service Discovery**.

Note: If **Enterprise Service Discovery** is not displayed, expand **Business Integration**, click **Enterprise Service Discovery** and then **Next**.

4. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for Flat Files (version 6.0.2) from the 'CWYFF_FlatFile' Connector Project** and click **Next**.
5. Complete the following steps in the Configure Settings for Discovery Agent window to enable data transformation.
 - a. In the **Folder name** field, type the path C:\Program Files\IBM\ResourceAdapters\FlatFiles\adapter\flatfile\samples. This tutorial will use the sample business object Customer.xsd located in this folder.
 - b. Leave the **Character Set** field empty.
 - c. From the **Content Type** list, select **text/xml** as the format to be used for all business objects. This is a one time setting and is used to bundle a content type with a corresponding data binding.
 - d. In the **DataBindingType** field, accept the default value, XMLBOSerializerDataBinding. This is the name of the data binding corresponding to the content type.

Note: The **DataBindingType** field is automatically populated based on the value selected for the **Content Type** field. If the value of the Content Type property is empty, the DataBindingType property does not display.

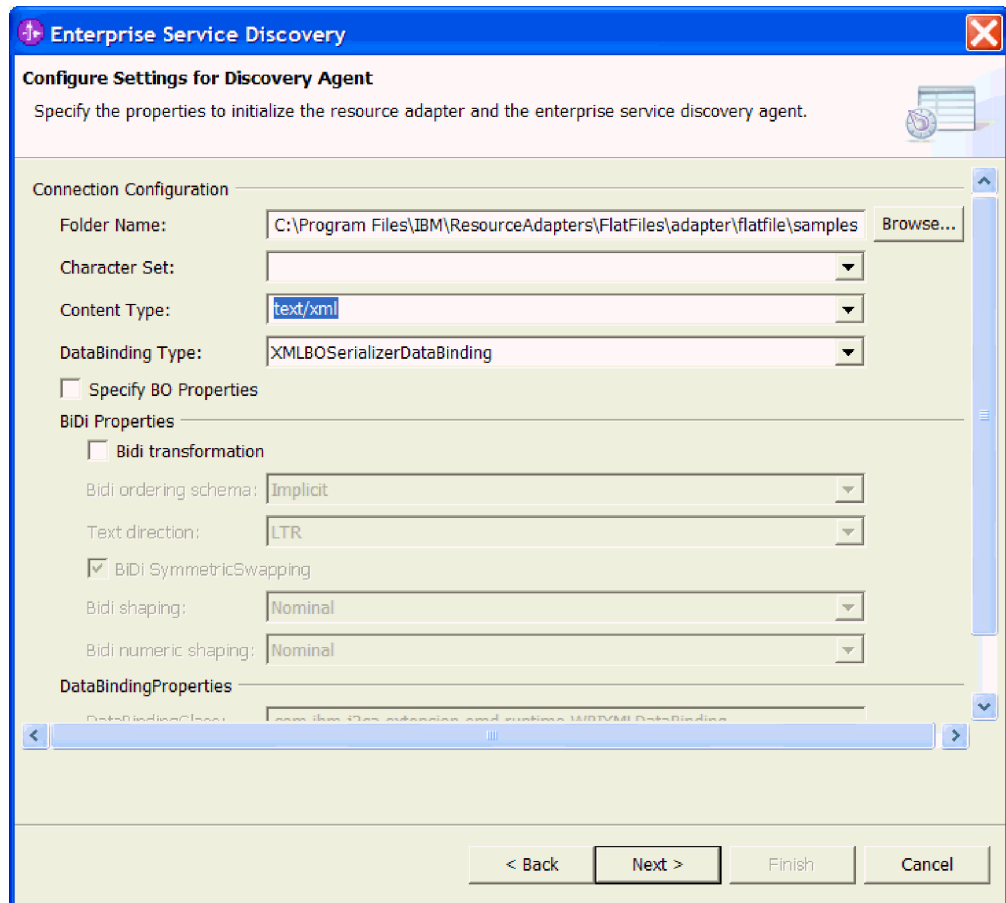


Figure 47. The Configure Settings for Discovery Agent window

6. Set the logging level so that you can see any errors that might arise during configuration.
 - a. Click **Show Advanced**. The button changes to **Hide Advanced**.
 - b. In the **Log file output location** field, accept the default path of the log file.
 - c. From the **Logging Level** list, select **FINEST**.
7. Click **Next**.

Result

The connection properties that the enterprise service discovery wizard uses to communicate with the enterprise information system are set.

Selecting the business objects and services to be used with the adapter

Use the enterprise service discovery wizard to browse for business objects and other metadata information located on the enterprise information system and select artifacts used for configuring inbound processing.

1. In the Find and Discover Enterprise Services window, click **Execute Query** to display the business objects found in the specified business objects folder.
2. In the "Objects discovered by query" pane, highlight the business objects you want to import, then click **Add to import list** to move the business objects to the "Objects to be imported" pane.

Note: To remove objects from the "Objects to be imported" pane, highlight the object that you want to remove and click **Remove**.

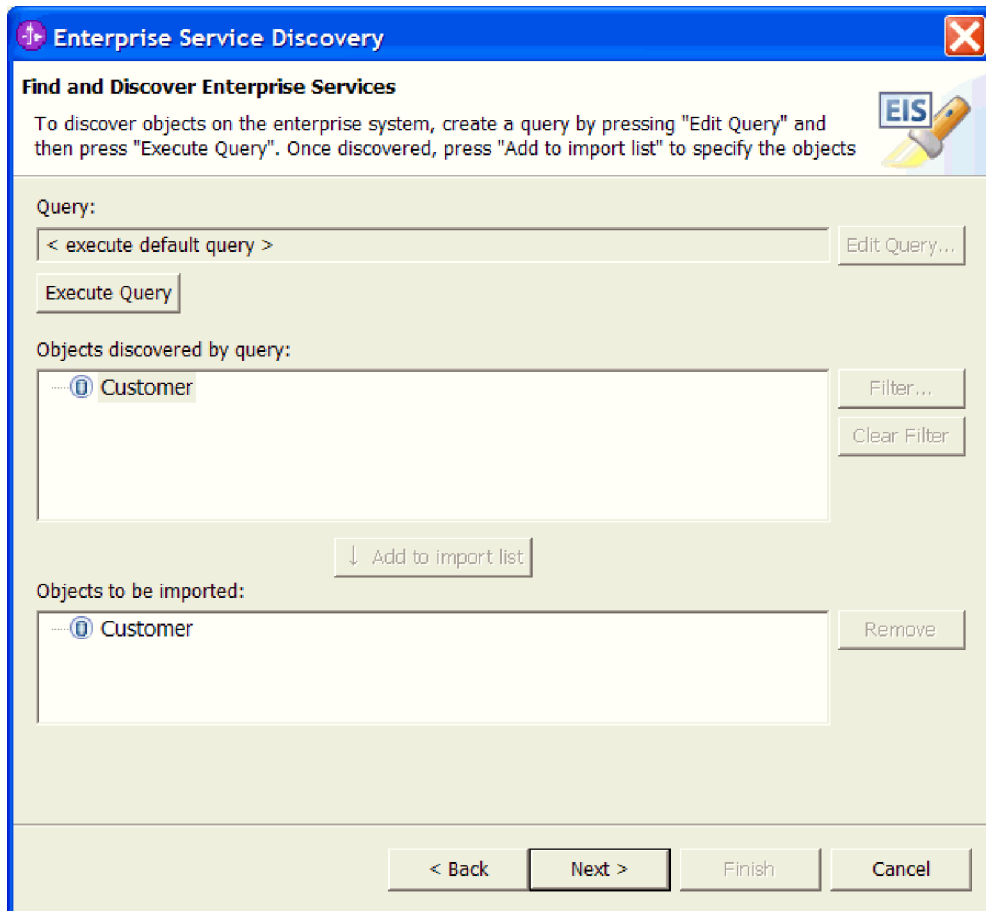


Figure 48. The Find and Discover Enterprise Services window

3. Click **Next**.

Result

You have selected the business objects or services that will be used to configure inbound event processing.

Configuring the selected objects

Once you have added business objects to the module, configure them for inbound operations.

1. In the Configure Objects window of the enterprise service discovery wizard, select **Inbound** from the **Service Type** list. The default base namespace for the business object schema to be generated is displayed. This value can be changed.
2. Type xsds in the **BO Location** field. This creates the specified directory to store your business objects called xsds in your connector project.
3. Click **Next**. All of the listed operations are selected by default. You can change the list by clicking the **Add** or **Remove** buttons.

Result

You have configured the objects that will be used with the inbound adapter project.

Generating artifacts

Define business object definitions and their related artifacts in WebSphere Integration Developer for use with your connector project.

1. To create a new business integration module, use the following procedure.
 - a. In the Generate Artifacts window, next to the **Module name** field, click **New**.
 - b. In the Integration Project window, select the default setting, **Create a module project**, and click **Next**.
 - c. In the New Module window, type "FlatFileInboundModule" in the **Module Name** field.
 - d. Under Module Location, select the **Use Default** check box.
 - e. Click **Finish**.
2. In the Generate Artifacts window, select the default setting, **Use discovered connection properties**.

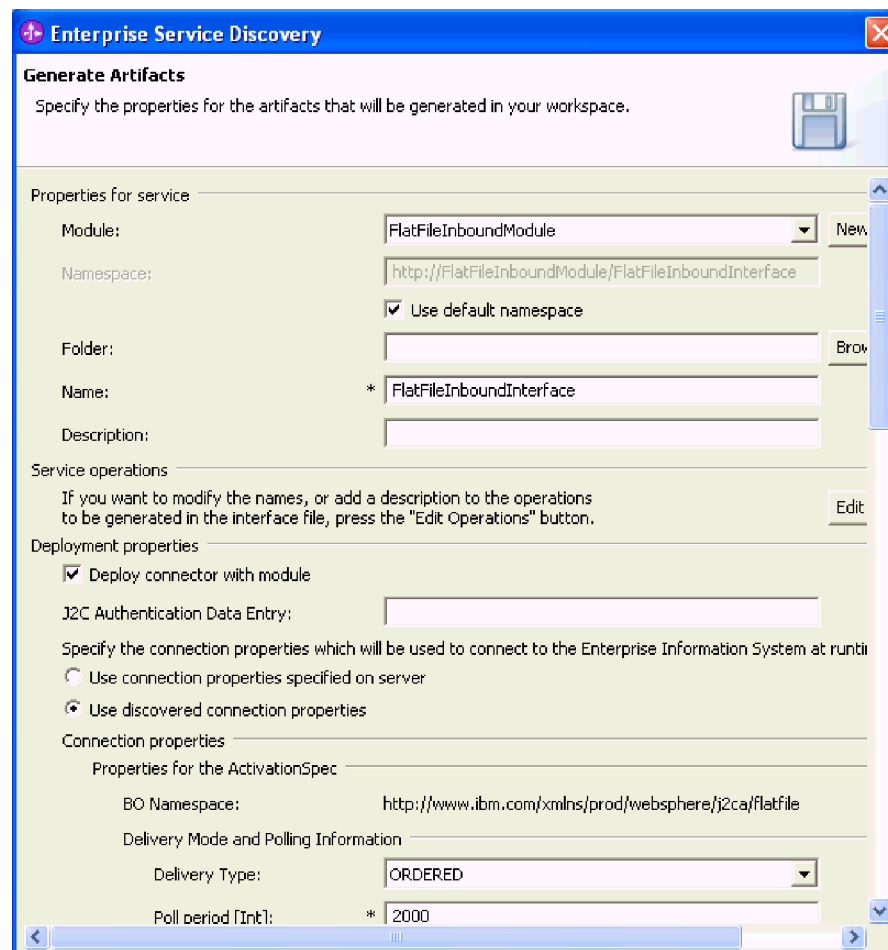


Figure 49. Generate Artifacts window, top section

3. Accept the default values for each of the required activation specification properties. Activation specification properties hold the inbound event processing configuration information for a message endpoint.

4. Type a value for each of the following required FlatFiles properties. These are activation specification properties that are specific to the adapter.
 - a. Uncheck the **Archiving Processed** check box.
 - b. In the **Event Directory** field, type the repository path name where the event files are stored. The event directory must be created on your local directory or the wizard will not be able to find inbound events.
 - c. In the **DataSourceJNDI Name** field, type the JNDI name of the data source created on WebSphere Process Server. For purposes of this tutorial, use jdbc/FFDB as the **DataSource JNDIName**.
 - d. In the **Event Table Name** field, type the name of the table used by the adapter for event persistence. For purposes of this tutorial, use FFDB as the **EventTableName**.
 - e. In the **EventContentType** field, type the mime type of the event file. For purposes of this tutorial, use text/xml as the **EventContentType**.

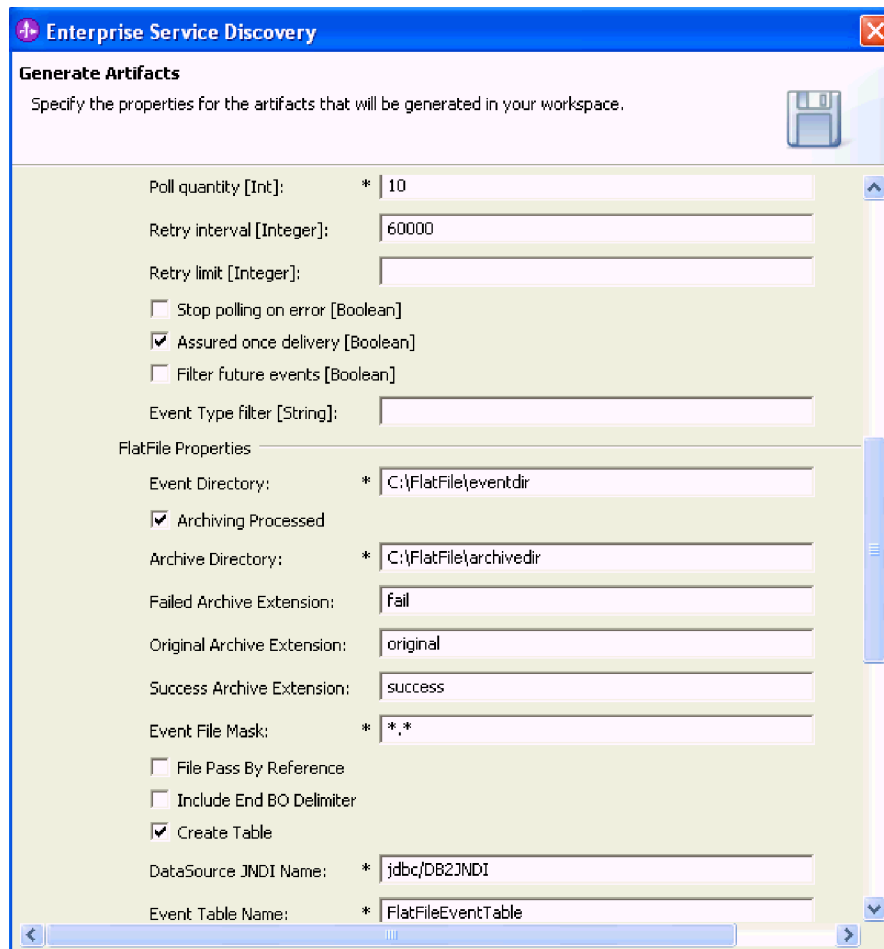


Figure 50. Lower section of the Generate Artifacts window

5. Click **Finish**.

Result

The FlatFileInboundInterface.wsdl and FlatFileInboundInterface.export artifacts, and the FlatFileBG, FlatFile, UnstructuredContent, CustomerWrapperBG, CustomerWrapper and Customer business objects are generated. The application

business objects specified by the user are updated with application-specific information for data transformation and saved in the business object location.

Generating reference bindings

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

Before you begin

An adapter project must be created and configured on your workspace. In addition, WebSphere Integration Developer must be running in order to complete this task.

About this task

Reference bindings bind the adapter project to the service component and are used by other WebSphere Business Integration SCA components to access the adapter. Creating a reference to the adapter from the project module links the adapter to the other server processes.

To generate reference bindings, use the following procedure.

How to perform this task

1. Expand FlatFileInboundModule and right-click the next level of FlatFileInboundModule. Select **Open With** → **Assembly Editor**. The Assembly Diagram window appears with an Import component in view.
2. In the Assembly Diagram window, hover over the icons and find **Component (with no implementation type)** (with no implementation type).

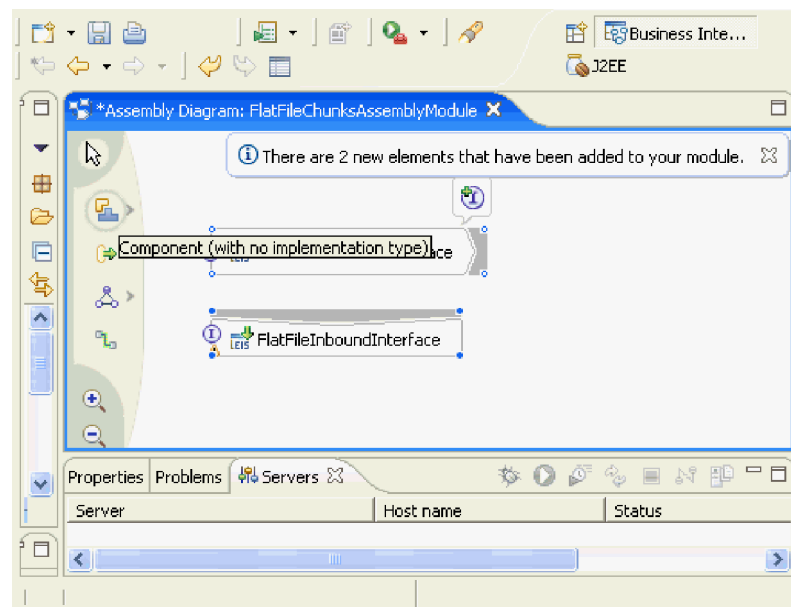


Figure 51. Component (with no implementation type) icon

3. Create a Component with no implementation type by single-clicking the icon and then selecting the second tier **Component (with no implementation type)** icon. Drag it into the Assembly Diagram window and single-click to place the

icon. To automatically organize the workspace, right-click in the Assembly Diagram window and select **Arrange Contents Automatically**.

4. To 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.

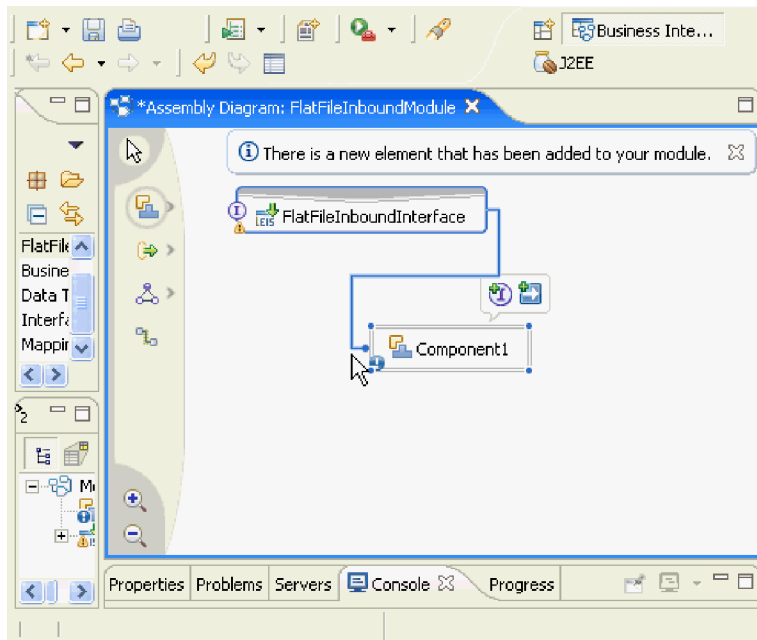


Figure 52. Adding a wire to the new component

5. 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 module's Import component.
6. Create a Java component that will act like an endpoint listener. During event delivery of the inbound operation, the adapter invokes the `emitFlatFile` 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 default package and click **OK**.
7. Click **File** → **Save**.

Result

You have just created reference binding for your adapter project.

What to do next

Use WebSphere Integration Developer to test the assembled adapter package.

Testing the assembled adapter application

Test the assembled `FlatFileInboundModule` using the WebSphere Integration Developer integration test client.

1. Switch to the Business Integration perspective in WebSphere Integration Developer.

2. Expand **FlatFileInboundModule**.
3. Right click **FlatFileInboundInterface** and select **Test** → **Attach**.
4. Click **Continue** to test the inbound module.
5. In the Select Deployment Location window, select **WebSphere Process Server v6.0** and click **Finish**.
6. Place some event files in the text format in the event directory specified earlier and verify that they are picked up by the adapter in the specified interval and delivered to the endpoint.

Result

You have successfully tested the "FlatFileModule" module and have verified that events have been delivered to the specified endpoint.

Tutorial 3: Splitting event files and reassembling with the append operation during inbound processing

In this scenario, the adapter splits a large file into chunks during inbound processing. When the business object containing the chunks reaches the endpoint, the chunks are reassembled using the append operation. This tutorial also demonstrates a pass through scenario where business objects contain unstructured records.

Creating the adapter project in WebSphere Integration Developer

Use WebSphere Integration Developer to create an inbound connector project and then import the resource adapter archive (RAR) file into the project. Importing the RAR file automatically creates a new J2EE connector project for the adapter in your workspace in WebSphere Integration Developer.

About this task

Note: If you have already created the adapter project, you do not need to do it again. You can skip these steps and begin configuring the adapter for inbound processing.

How to perform this task

1. In WebSphere Integration Developer, switch to the J2EE perspective:
 - a. Click **Window** → **Open Perspective** → **Other**.
 - b. Click **J2EE**.
If **J2EE** is not displayed, select the **Show all** check box, click **J2EE**, and click **OK**.
 - c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.
 - d. Click **OK**.
2. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

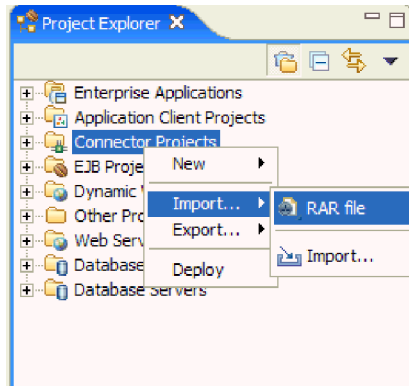


Figure 53. Importing the RAR file

3. Find the RAR file on your local file system by clicking **Browse** and navigating to the directory in which Adapter for Flat Files was installed. If you chose the default path when installing the adapter, use the following directory path:
C:\Program Files\IBM\ResourceAdapters\FlatFile\adapter\FlatFile\deploy.
4. Accept the default setting (**CWYFF_FlatFile**) for **Connector project**.
The connector project has the same name as the RAR file.

Note: If a project named CWYFF_FlatFile already exists in this workspace, the name in the Connector project field has a number appended to it (for example, CWYFF_FlatFile1).

5. Accept the default value in the **Target server** field.
The default value is the test environment for WebSphere Process Server, which is installed as part of WebSphere Integration Developer.
6. Clear the **Add module to an EAR project** check box.

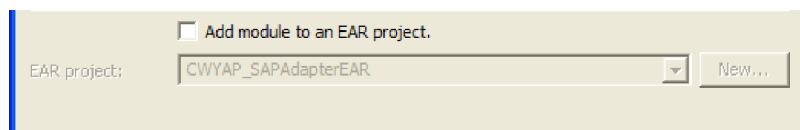


Figure 54. Clearing the Add module to an EAR project check box

Notice that the EAR project field becomes unavailable after you remove the check mark.

7. Click **Finish**.

Result

A new J2EE connector project, named CWYFF_FlatFile, is created in your WebSphere Integration Developer workspace.

Configuring the adapter for inbound processing

Use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties, select business objects or services that are in the enterprise information system, and generate business object definitions and related artifacts for inbound processing.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to set the connection properties necessary to enable communication with the enterprise information file system. Once communication is established between the two entities, the enterprise service discovery wizard can obtain the necessary metadata from the enterprise information file system.

1. Optional: Switch to the Business Integration Perspective by clicking **Window** → **Open** → **Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
2. Start the enterprise service discovery wizard. Click **File** → **New** → **Enterprise Service Discovery**, and then click **Next**.

Note: If **Enterprise Service Discovery** is not displayed, click the **Show all wizards** check box, expand **Business Integration**, click **Enterprise Service Discovery**, and then click **Next**.

3. In the Select an Enterprise Service Resource Adapter window, select **IBM WebSphere Adapter for Flat Files (version 6.0.2) from the 'CWYFF_FlatFile' Connector Project** and click **Next**.
4. In the Configure Settings for Discovery Agent window, click **Next**.

Result

The connection properties that the enterprise service discovery wizard uses to communicate with the enterprise information system are set.

Selecting the business objects and services to be used with the adapter

Use the enterprise service discovery wizard to query business objects and metadata information on the enterprise information system. Then, select the artifacts used to configure the project for inbound processing.

1. In the Find and Discover Enterprise Services window, click **Execute Query** to display the business objects for the adapter.
2. In the "Objects discovered by query" pane, highlight the **UnstructuredData** business objects you want to import, then click **Add to import list** to move the business objects to the "Objects to be imported" pane.

Note: To remove objects from the "Objects to be imported" pane, highlight the object that you want to remove and click **Remove**.

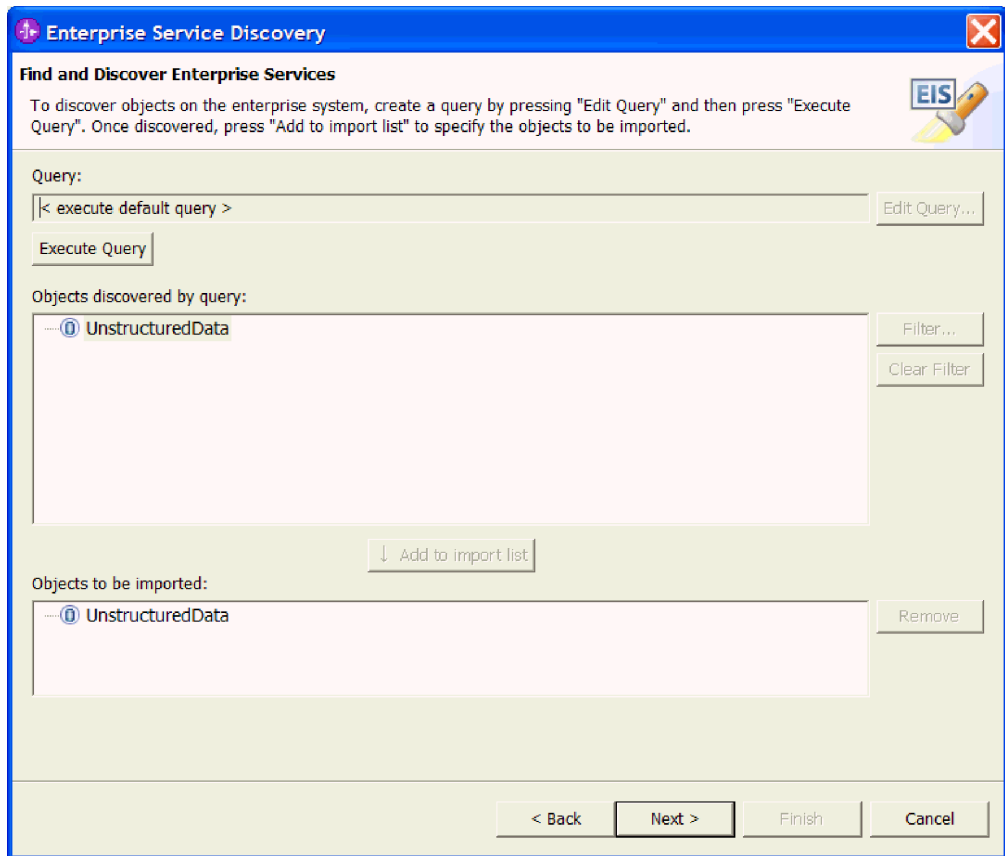


Figure 55. The Find and Discover Enterprise Services window

3. Click **Next**.

Result

You have selected the business objects or services that will be used to configure inbound event processing.

Configuring the selected objects

Once you have added business objects to the module, configure them for inbound operations.

1. In the Configure Objects window of the enterprise service discovery wizard, select **Inbound** from the **Service Type** list. The default base namespace for the business object schema to be generated is displayed. This value can be changed.
2. Type xsds in the **BO Location** field. This creates the specified directory to store your business objects called xsds in your connector project.
3. Click **Next**. All of the listed operations are selected by default. You can change the list by clicking the **Add** or **Remove** buttons.

Result

You have configured the objects that will be used with the inbound adapter project.

Generating artifacts

Define business object definitions and their related artifacts in WebSphere Integration Developer for use with your connector project.

1. To create a new business integration module, use the following procedure.
 - a. In the Generate Artifacts window, next to the **Module name** field, click **New**.
 - b. In the Integration Project window, select the default setting, **Create a module project**, and click **Next**.
 - c. In the New Module window, type FlatFileChunksAssemblyModule in the **Module Name** field.
 - d. Under Module Location, select the **Use Default** check box.
 - e. Click **Finish**.
2. In the Generate Artifacts window, select the **Deploy connector with module** check box.
3. Select the default setting, **Use discovered connection properties**.
4. Type the value for each of the required Flat Files properties.

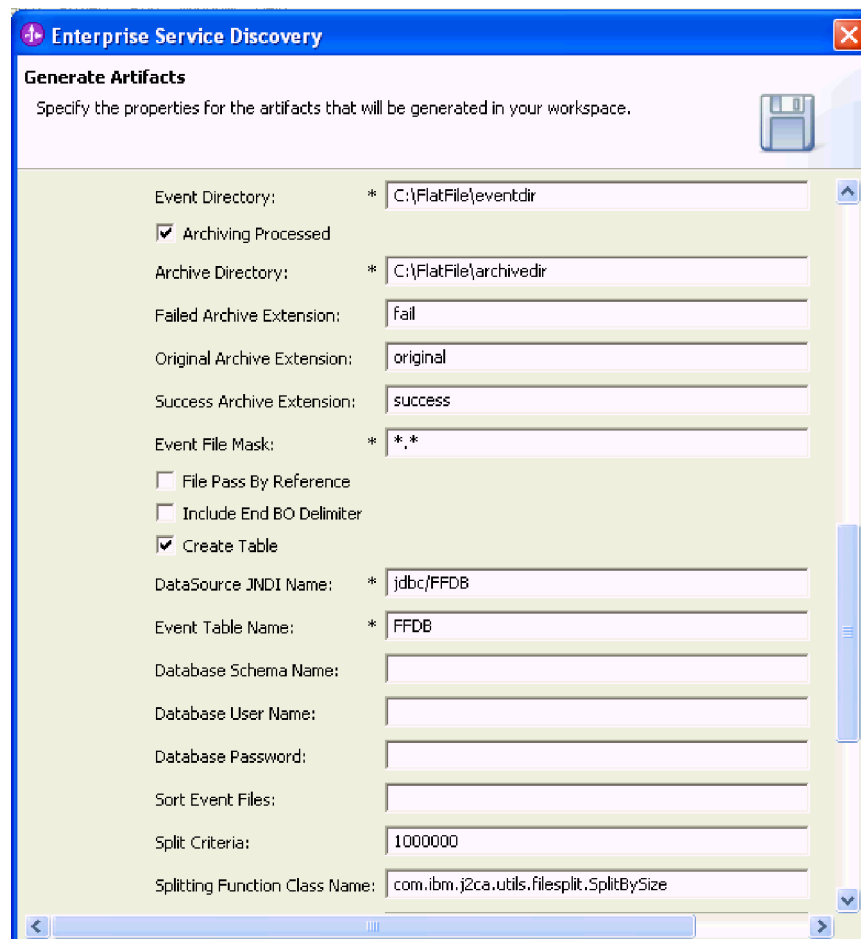


Figure 56. Generate Artifacts window

- a. In the **Event Directory** field, type the repository path name where the event files are stored. The event directory must be created on your local directory or the wizard will not be able to find inbound events.
- b. In the **DataSourceJNDI Name** field, type the JNDI name of the data source created on WebSphere Process Server. For purposes of this tutorial, use jdbc/FFDB as the **DataSource JNDI Name**.

- c. In the **Event Table Name** field, type the name of the table used by the adapter for event persistence. For purposes of this tutorial, use FFDB as the **EventTableName**.
 - d. In the **Split Criteria** field, type a value to split event files based on either size or a delimiter. For purposes of this tutorial, use 1000000.
 - e. In the **Splitting Function Class Name** field, type `com.ibm.j2ca.utils.filesplit.SplitBySize`. This class used to split files based on size.
5. Click **Finish**.

Result

A new business integration module has been created and configured for inbound processing.

Generating the FlatFileOutboundModule used for testing

Use WebSphere Integration Developer to create an outbound interface in the FlatFileChunksAssembly module. The outbound interface is used to execute the create and append operations needed to complete this scenario.

1. Create the outbound interface by following the same steps used to create the inbound interface in these sections:
 - a. Setting connection properties for enterprise service discovery
 - b. Selecting the business objects and services
2. In the Configure Objects window of the enterprise service discovery wizard, select **Outbound** from the **Service Type** list. The default base namespace for the business object schema to be generated is displayed. This value can be changed.
3. Type the location of the business object in the **BO Location** field. This creates the specified directory name in your connector project.
4. Click **Next**. All of the listed operations are selected by default. You can change the list by clicking the **Add** or **Remove** buttons.
5. In the Generate Artifacts window, select the **Deploy connector with module** check box.
6. Select the default setting, **Use discovered connection properties**.
7. In the Generate Artifacts window, specify the Output Directory where all of the split files can be assembled and placed.

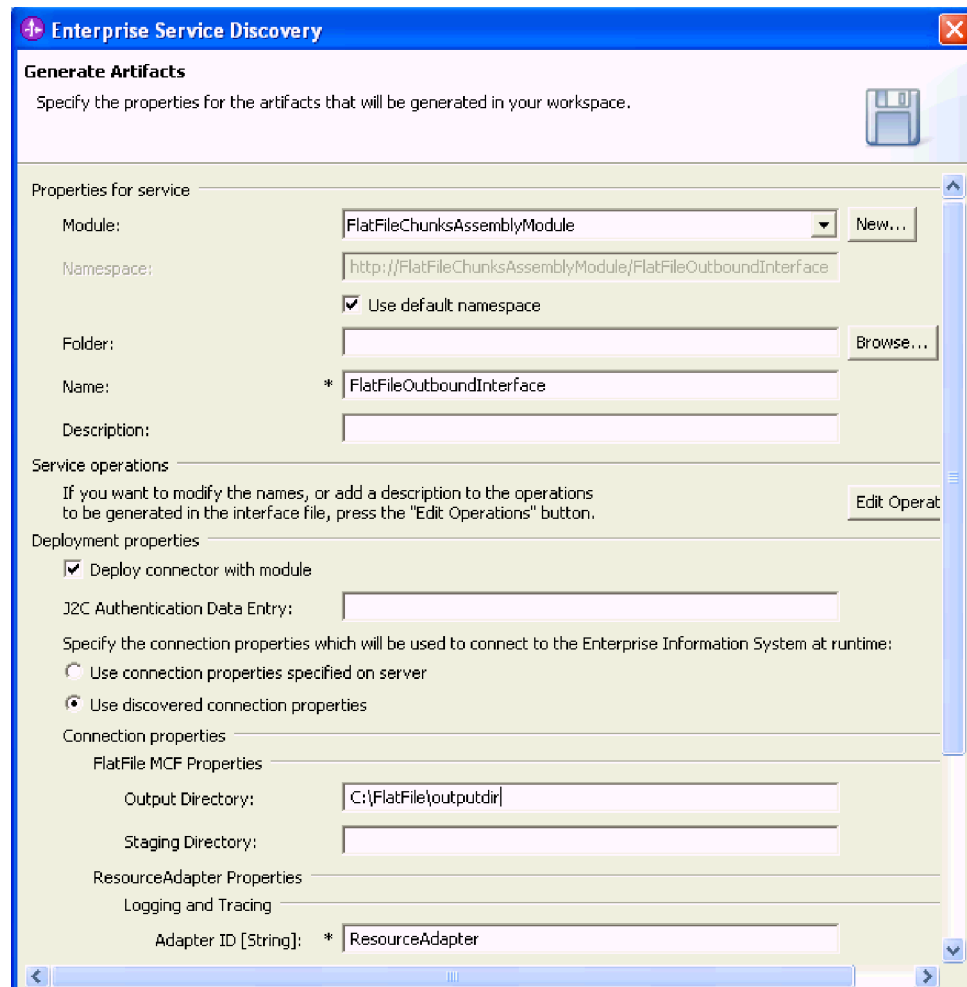


Figure 57. Generate Artifacts window

8. Optional: Specify a Staging Directory where the split files can be created before moving to them output directory.
9. Click **Finish**.

Result

The FlatFileOutboundInterface is created in the FlatFileChunksAssembly Module.

Generating reference bindings

Use WebSphere Integration Developer to generate reference bindings that are used by other WebSphere Business Integration SCA components to access the adapter.

Before you begin

An adapter project must be created and configured on your workspace. In addition, WebSphere Integration Developer must be running in order to complete this task.

About this task

Create a reference to the adapter from the project module to link the adapter to other server processes.

To generate reference bindings, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
2. Right-click the FlatFileChunksAssemblyModule, and select **Open With** → **Assembly Editor**. The Assembly Diagram window appears with the modules Import component in view.

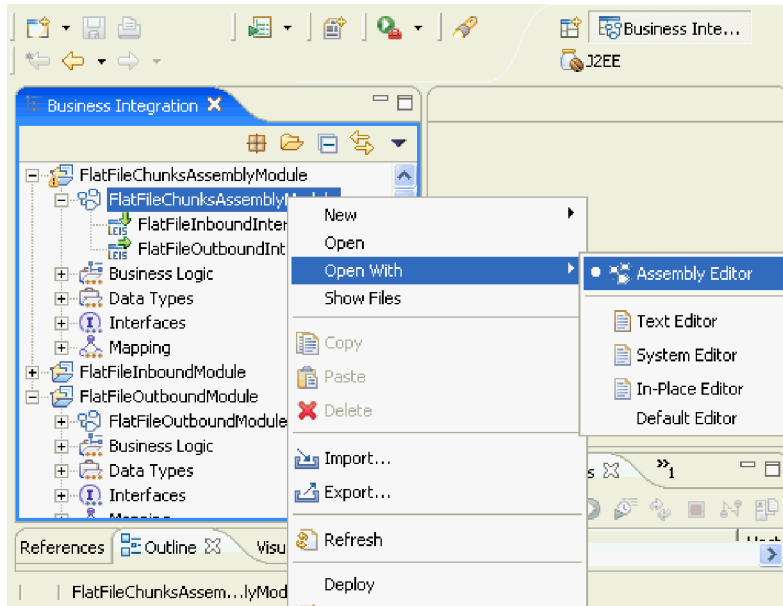


Figure 58. Assembly Editor selection

3. In the Assembly Diagram window, create a new component by clicking the top most icon in the left-side (vertical) frame of the Assembly Diagram Window. A new menu of icons appears.

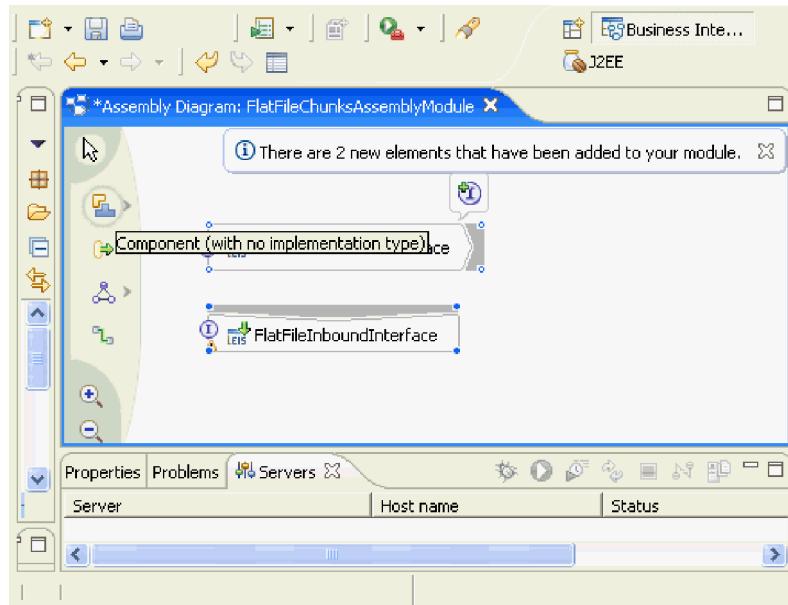


Figure 59. Component (with no implementation type) icon

4. Create a Component with no implementation by dragging the Create a Component icon into the Assembly Diagram window. To automatically organize the workspace, right-click in the Assembly Diagram window and select **Arrange Contents Automatically**.
5. Create a wire from the FlatFileInboundInterface to the Component1 by clicking and dragging the module's Import component to Component1. This draws a wire from the FlatFileInboundInterface to Component1.

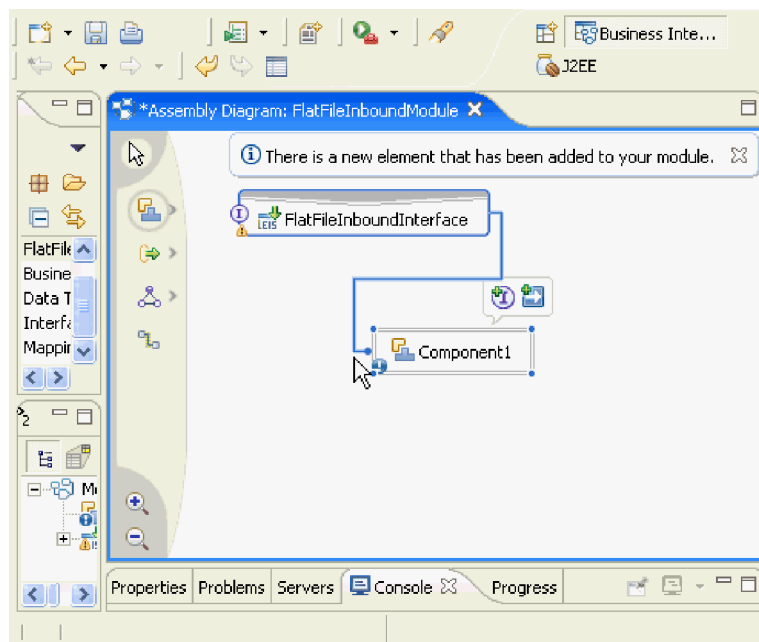


Figure 60. Creating a wire from FlatFileInboundInterface to Component1

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 wire from Component1 to the FlatFileOutboundInterface. This enables Component1 to make calls to the FlatFileOutboundInterface.

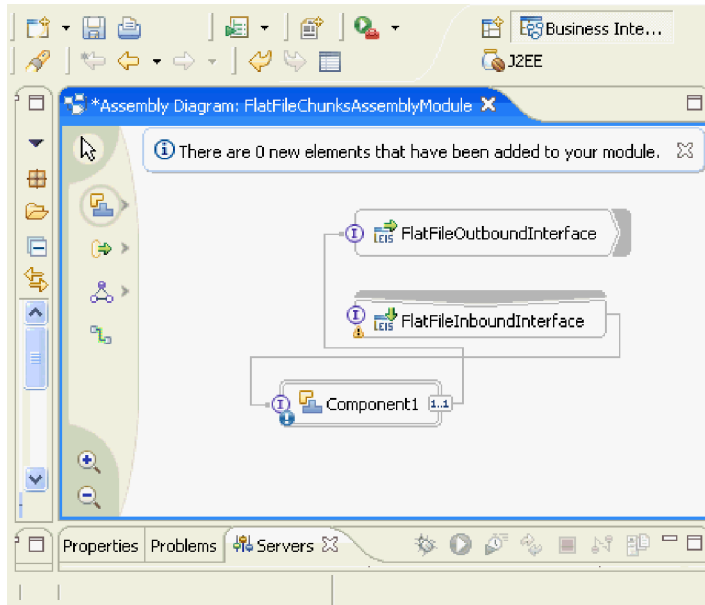


Figure 61. Creating a wire from Component1 to FlatFileOutboundInterface

8. Create a Java component that will act like an endpoint listener. During event delivery of the inbound operation, the adapter invokes the emitFlatFile method of the component implementation and passes the inbound business object as a parameter.
 - a. Right click Component1 and select **Generate implementation** → **Java**.
 - b. Select the default package and click **OK**.
 - c. In the generated Java implementation, scroll to the emitFlatFile method.
9. Click **File** → **Save**.

Result

A reference binding is created for your adapter project.

What to do next

Use WebSphere Integration Developer to test the assembled adapter package.

Testing the assembled adapter application

Test the assembled FlatFileChunkAssemblyModule using the WebSphere Integration Developer integration test client.

1. Switch to the Business Integration perspective in WebSphere Integration Developer.
2. In WebSphere Integration Developer, right click **FlatFileChunksAssemblyModule** and select **Test** → **Attach**.
3. Click **Continue** to test the inbound module.

4. In the Select Deployment Location window, select **WebSphere Process Server v6.0** and click **Finish**.
5. Place an event file that is larger than 1000000 bytes in the event specified directory.
6. Verify that a complete file is created in the C:\flatfile\outputdir directory.

Result

The inbound module of the adapter will split the file into chunks and deliver it to the specified directory. The outbound module of the adapter will reassemble the chunks in to one file.

Troubleshooting the tutorial

If you are unable to deploy the module using WebSphere Integration Developer, use the administrative console of WebSphere Process Server.

1. From WebSphere Integration Developer, switch to the J2EE perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **J2EE** and click **OK**.
2. In the Project Explorer pane, expand **Enterprise Applications**.
3. Right-click the assembled adapter module, then click **Export** → **EAR file**.
4. In the EAR Export window, enter the following information:
 - a. From the **EAR project** list, select the name of the assembled adapter application.
 - b. From the **Destination** list, either select or browse to the exact location of the EAR file.
 - c. Select the **Export source files** check box.
 - d. Select the **Overwrite existing file** check box.
 - e. Select the **Include project build paths and meta-data files** check box.
 - f. Click **Finish**.
5. To start WebSphere Process Server in WebSphere Integration Developer, use the following procedure:
 - a. In WebSphere Integration Developer, open the Servers view by clicking **Window** → **Show View** → **Servers**.
 - b. To start WebSphere Process Server, right-click **WebSphere Process Server v6.0** and click **Start**.
6. Once the server is started, right-click **WebSphere Process Server v6.0**, then select **Run administrative console**.
7. In the WebSphere Administrative Console Login page, enter a User ID, then click **Log in**.
8. To Install the EAR file to WebSphere Process Server, use the following procedure:
 - a. In the administrative console, expand **Applications** and select **Install New Applications**.
 - b. In the "Enterprise Applications" page, select the EAR file from the list of enterprise applications and click **Install**.
 - c. In the "Preparing for the application installation" page, select **Local file system**, then click **Browse** to navigate to the exact location of the EAR file.
 - d. Click **Next**.
 - e. In the "Choose to generate default mappings and bindings" page, click **Next**.

f. In the "Install New Application" page, click **Step 9**.

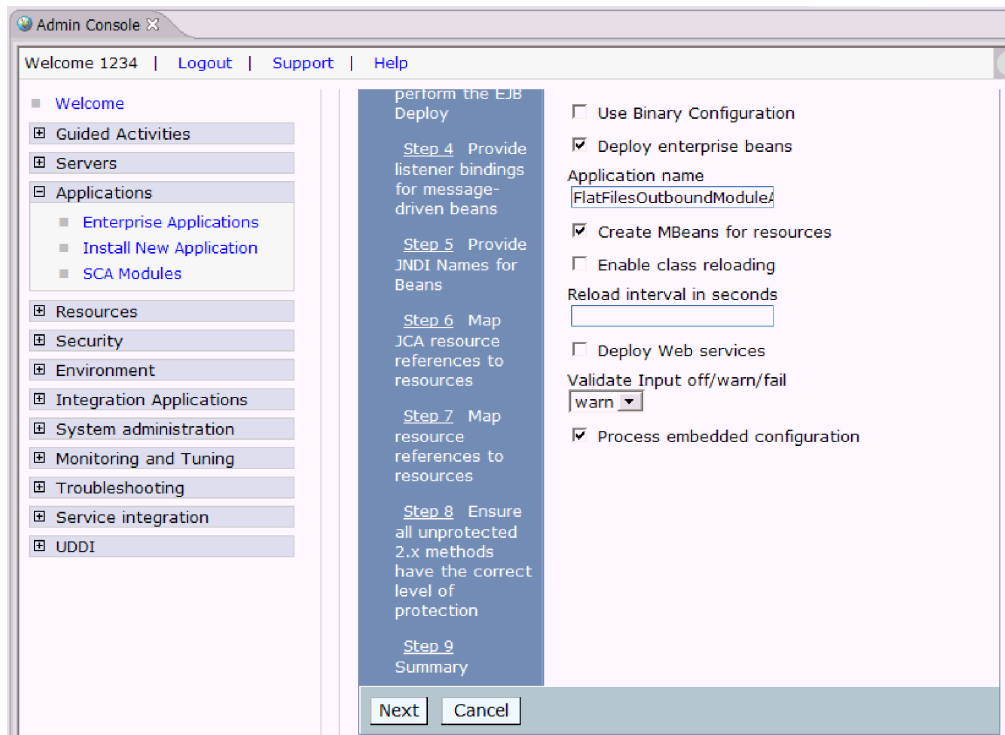


Figure 62. The Install New Application page

- g. In the "Summary" page, verify that all options are correct and click **Finish**. When the adapter project is successfully installed, a confirmation of a successful installation is displayed.
- h. Click the **Save to Master Configuration** link that appears at the end of the list of installation messages.
- i. In the Enterprise Applications > Save page, click **Save**.

Chapter 11. Reference information

Reference information supports the tasks you want to complete. The information includes all configurable properties for the Adapter for Flat Files, messages, and related product information.

Enterprise service discovery connection properties

Enterprise service discovery connection properties include outbound and inbound connection properties required for performing metadata discovery configuration. You configure these properties using the enterprise service discovery wizard when you initially deploy the adapter.

When you run the enterprise service discovery wizard in WebSphere Integration Developer, specify the connection properties listed below.

Table 9. Connection configuration properties for the Adapter for Flat Files

Property	Type	Description	Default value	Required
Folder Name	WBIFolder Property	Folder where XSD files are located. If the XSD files are not present in this folder, then the adapter does not perform any data transformations. If this property is not specified, the enterprise service discovery wizard treats the scenario as a pass through scenario without data transformation.	Empty	No

Table 9. Connection configuration properties for the Adapter for Flat Files (continued)

Property	Type	Description	Default value	Required
Character Set	String	<p>List of encodings supported by the Java Virtual Machine. The Character Set property sets the encoding to be used for business objects.</p> <p>If the content of the file are in English, you do not need to select a Character Set. However, if the file content has a particular encoding, then you have to select that particular value from the Character Set so that the adapter will use that particular encoding while writing to the file system.</p>	Empty	No
Content Type	List of Values	<p>Format to be used for all business objects. For example, text/xml. Includes formats supported by the existing data handlers. You can edit this list of values. If the Content Type property value is empty, the enterprise service discovery wizard treats the scenario as a pass through scenario without data transformation.</p>	Empty	No
DataBinding Type	List of Values	<p>Name of the data binding corresponding to the content type. For example, XMLBOSerializer DataBinding. You can edit this list of values. If the value of the Content Type property is empty, the DataBindingType property does not display.</p>	Empty	No

Table 9. Connection configuration properties for the Adapter for Flat Files (continued)

Property	Type	Description	Default value	Required
DataBinding Properties	Property Group	Property group for the selected data binding type. This list of properties depends on the value set in the DataBinding Type field.	Null	No
Specify BO Properties	Boolean	Indicates whether or not you want to specify properties for individual business objects.	False	No
Bidi transformation	Boolean	Enables the adapter to exchange bidirectional data between WebSphere Process Server components and the enterprise information system.	False	No
Log file output location	String	Absolute path of the log file generated by the enterprise service discovery wizard.		
Logging Level	List of values	The level of logging used by the adapter. Levels include: Severe, warning, Info, Config, and Finest.	Severe	No

Adapter configuration properties

The properties in this section should be configured using the enterprise service discovery wizard before deployment or with the administrative console after deployment.

Resource adapter properties

Resource adapter properties consist of logging and tracing, bidirectional language support, and activities specific to the adapter, such as the default configuration properties of the adapter. You configure these properties using the WebSphere Process Server administrative console.

When you configure the adapter, specify the resource adapter properties listed in the Resource adapter properties for the Adapter for Flat Files table below.

Table 10. Resource adapter properties for the Adapter for Flat Files

Property	Type	Description	Globalized	Default Value	Required
Adapter ID	String	Identifies a deployment instance of the adapter. If you are using more than one instance of an adapter, the value of this property must be unique.	Yes		Yes
enableHASupport	String	When the enableHASupport property is set to true, only one of the replicated adapter instances actively polls for events while other instances are in standby mode. If the enableHASupport property is set to false, all of the adapter instances replicated on cluster members actively poll for events. This may result in event duplication. Do not change the value of enableHASupport to false for single server environments.		True	No
Log file name	String	Full path of the log file.	Yes		Yes
Log files	Integer	Number of log files to use. When a log file reaches its maximum size, the adapter uses another log file.	Yes	1	No
Log file max size	Integer	Size of the log files in kilobytes. If no value is specified, then the file does not have a maximum size.	Yes	0	No
Trace file name	String	Full path of the trace file.	Yes		No

Table 10. Resource adapter properties for the Adapter for Flat Files (continued)

Property	Type	Description	Globalized	Default Value	Required
Trace files	Integer	Number of trace files to use. When a trace file reaches its maximum size, the adapter uses another trace file.	Yes	1	No
Trace file max size	Integer	Size of the trace files in kilobytes. If no value is specified, then the file does not have a maximum size.	Yes	0	No

Managed (J2C) connection factory properties

Managed connection factory configuration properties are used at run time to create an outbound connection instance with an enterprise information system.

When you configure the adapter, specify the properties listed below.

Note: The enterprise service discovery wizard refers to these properties as managed connection properties and WebSphere Process Server refers to these as (J2C) connection factory properties.

Table 11. Managed (J2C) connection factory properties

Property	Type	Description	Globalized	Bidirectional Transport Support	Default Value	Required
Output Directory	String	Absolute path of the directory accessed for outbound operations.	Yes	Yes	None	No
Staging Directory	String	If this property is specified, a file is written first to the staging directory before it is written to the output directory.	Yes	Yes	None	No
OutputFileName	String	If this property is specified, the file created in the output directory will have this name. (This value will be overridden if a value for OutputFileName is specified in the record object.)	Yes	Yes	None	No

Activation specification properties

Activation specification properties hold the inbound event processing configuration information for a message endpoint. They can be set through the enterprise service discovery wizard or the administrative console.

When you configure the adapter, specify the activation specification properties listed below.

Table 12. Activation specification properties

Property	Type	Description	Bidirectional Transport Support	Default Value	Required
Archive Directory	String	Directory where the adapter archives processed event files.	Yes	None	Yes, if Archiving Processed = true
ArchivingProcessed	Boolean	Determines if the adapter should archive processed events.	Yes	True	No
AssuredOnceDelivery	Boolean	Provides assured once only event delivery. If the value is true, the adapter attempts to store XID information in the event store. If the value is false, the adapter will not attempt to store XID information in the Event Store. Note: Setting the value of this property to false improves adapter performance.	No	True	Yes
Default Object Name	String	Name of the business graph that is used by the Data Transformation Framework before delivering an event to an endpoint. For example, the generic wrapper business graph used by the adapter is FlatFileBG.	No	None	No
CreateTable	Boolean	Determines if the event persistence table gets created automatically or manually. If the value is true and the event persistence table does not exist, then the adapter creates the table. If the value is false, then the adapter does not create the table and the user must manually create the table.	No	True	No
DatabasePassword	String	Password used by event persistence to obtain the JDBC database connection from the data source.	Yes	None	No
DatabaseSchemaName	String	Schema name of the database used by event persistence.	Yes	None	No
DatabaseUsername	String	User name used by event persistence to obtain the JDBC database connection from the data source.	Yes	None	No

Table 12. Activation specification properties (continued)

Property	Type	Description	Bidirectional Transport Support	Default Value	Required
DataSourceJNDIName	String	JNDI name of the data source used by event persistence to obtain the JDBC database connection. The data source must be created in WebSphere Process Server.	Yes	None	Yes
EventTableName	String	Name of the table used by the adapter for event persistence. This value must be unique for each activation specification instance when multiple activation specification instances are used.	Yes	None	Yes
Event Content Type	String	Represents the mime type of the event file.	No	None	No
Event Directory	String	Directory where the event files are stored by the back-end enterprise information system.	Yes	None	Yes
EventFileMask	String	Specifies the filter for the event files. The file filter is a well-qualified valid regular expression consisting of alphanumeric characters and only the wildcard "**".	Yes	*.*	Yes
EventTypeFilter	String	List of event types that the adapter can deliver. The adapter will not deliver events that are not in the list. If the list is empty (null), no filter is applied, and all events are delivered.	No	Null	No
Failed Archive Extension	String	Specifies the file extension used to archive unsuccessfully processed business objects in the input event file.	Yes	fail	No
File Content Encoding	String	Determines how files are read and written by the adapter.	No	None	No
File Pass By Reference	Boolean	Determines if the adapter needs to load the contents of a file or just provide information about the directory name and the file name to the endpoint.	No	false	Yes
FilterFutureEvents	Boolean	If true, the adapter will inspect the timestamp on each event and compare it to the system time. If the event time is ahead of the system time, the event will not be delivered.	No	false	Yes

Table 12. Activation specification properties (continued)

Property	Type	Description	Bidirectional Transport Support	Default Value	Required
Include End BO Delimiter	Boolean	When this property is set to true, the delimiter value specified in the SplitCriteria property is also sent with the business object content for further processing. Note: This property is valid only if event file splitting is based on a delimiter.	No	false	No
Original Archive Extension	String	Specifies the file extension used to archive the original event file. This property also preserves the entire event file for reference in case any of the business objects fail processing.	No	original	Yes
PollPeriod	Integer	Rate, in milliseconds, which the adapter polls the enterprise information system event store for new inbound events. The poll cycle is established at a fixed-rate. If an the poll cycle is delayed for any reason, for example, the poll cycle takes longer than expected to complete, then the next cycle immediately starts. If the value of this property is set to 0, the adapter will not wait between cycles to begin polling.	No	500	Yes
PollQuantity	Integer	Determines the number of events to deliver to each endpoint per poll cycle.	No	1	Yes
Retry Interval	Integer	Determines the interval of time, in milliseconds, that an adapter uses to reconnect to the enterprise information system after the adapter encounters an inbound connection error.	No	60000 (1 minute)	No
Retry Limit	Integer	Specifies the number of times the adapter attempts to reconnect to the enterprise information system before the adapter stops trying. A value of 0 means that the adapter attempts re-connection an unlimited number of times.	No	0 (unlimited)	No

Table 12. Activation specification properties (continued)

Property	Type	Description	Bidirectional Transport Support	Default Value	Required
Sort Event Files	String	<p>Determines the sorting order of polled event files. The values supported include:</p> <ul style="list-style-type: none"> • Filename - sort in ascending order on file name • Timestamp - sort in ascending order on last modified timestamp • <Blank> - not sorted <p>To support globalization, the sorting of file names is provided according to the system locale. The ICU4J package is used to track the locales and the rules corresponding to the locales.</p>	No	<blank> (= not sorted)	No
Split Criteria	String	<p>This attribute takes different values based on the value set in the SplittingFunctionClass Name property.</p> <ul style="list-style-type: none"> • If the SplittingFunctionClass Name property is set to a value which splits based on a delimiter, then the SplitCriteria property must contain the delimiter which separates the business objects in the event file. • If the SplittingFunctionClass Name property is set to a value that splits based on size, then the SplitCriteria property must contain a valid number that represents the size in bytes. If the event file size is greater than this value, it is split into chunks of this value and that number of chunks are posted. If the event file size is less than this value, the entire event file is posted. <p>When the SplitCriteria property value is set to zero, file splitting is disabled.</p> <p>Note: During an inbound pass through scenario, if file splitting is based on size, and FilePassByReference property is enabled, then the event files are not split into chunks.</p>	Yes	0	No

Table 12. Activation specification properties (continued)

Property	Type	Description	Bidirectional Transport Support	Default Value	Required
Splitting Function Class Name	String	<p>Uses the fully qualified class name in order to split event files. This property takes either of the following two class values:</p> <ul style="list-style-type: none"> com.ibm.j2ca.extension.utils.filesplit.SplitByDelimiter - A class which splits the event file based on a delimiter that separates business objects in the event file com.ibm.j2ca.utils.filesplit.SplitBySize - A class which splits event files based on the size of the event file. <p>The delimiter or file size is set in the SplitCriteria property. Note: If the EventContentType property is null, then this property is automatically set to the class name that splits files based on size.</p>	No	com.ibm.j2ca.utils.filesplit.SplitBySize	No
Stop polling on error	Boolean	Stops the adapter when an error is encountered during polling.	No	false	No
Success Archive Extension	String	Specifies the file extension used to archive all the successfully processed business objects.	No	success	Yes

FlatFileActivationSpec properties are the properties that are applicable only to the 6.0.1 version of the Adapter for Flat Files. These properties are necessary for backward compatibility and migration, where the adapter will support two activation specifications. However, there is no scenario to run the enterprise service discovery wizard for the 6.0.1 version properties for inbound processing. Rather, running the enterprise service discovery wizard for inbound processing is with respect to 6.0.2 configuration only.

Table 13. FlatFileActivationSpec properties

Property	Type	Description	Bidirectional Transport Support	Default Value	Required
AutocreateEDT		If the value of this property is set true, an in-memory table is created and the event management framework uses this table to track event delivery.	No	True	Yes
EDTDatabaseName		Name of the database used by the event management framework.	No	None	Yes
EDTDriverName		Name of the driver for the EDT database.	No	None	Yes

Table 13. FlatFileActivationSpec properties (continued)

Property	Type	Description	Bidirectional Transport Support	Default Value	Required
EDTTableName		Name of the table on the EDT database used by the event management framework.	No	None	Yes
EDTUserName		User ID credential to access the EDT DB	No	None	Yes
EDTUserPassword		Password credential to access the EDT DB	No	None	Yes
FFEventTableName		Name of the table that will be utilized by the adapter for tracking event status values.	Yes	FFLOG	Yes
FFDatabaseName		This is the database that is used internally by the adapter for inbound operations.	Yes	FFDB	Yes
FileChunkSize		Specifies the size of each chunk in bytes, when the file needs to be split into chunks. When the property FileSplitThreshold is empty, this property is disabled.	Yes	8000	No
FileSplitThreshold		Specifies the threshold file size for splitting files in bytes. All files above this file size are split into chunks. When this attribute is empty, file splitting is disabled.	Yes	-1	No

Adding jar files to WebSphere Integration Developer versions 6.0.1.1 and earlier

If you are using WebSphere Integration Developer version 6.0.1.1 or earlier, you must manually add three jar files to the classpath of the connector project.

You must have installed the adapter and all of the adapter prerequisites before the jar files can be added to the connector project in WebSphere Integration Developer.

1. Open WebSphere Integration Developer.
2. In J2EE perspective, right-click the connector project and select **Properties**.
3. Select **Java Build Path** and click **Add External Jars**.
4. Select your WebSphere Process Server or Enterprise Server Bus Install/lib folder and select `ffdcSupport.jar`, `aspectjrt.jar` and `icu4j_3_2.jar`.
5. Click **Open** and then **OK**.

Settings for controlling bidirectional transformation

Within each category of adapter properties, certain properties can be set to control bidirectional transformation of content or metadata. Properties controlling bidirectional transformation can be set for the resource adapter, the managed connection factory, and the activation specification; data transformation properties can also be set to control bidirectional transformation.

Resource adapter properties

The following resource adapter properties can be set to control bidirectional transformation.

- EIS BiDi Format
- Metadata BiDi Format
- Skip BiDi Transformation
- EIS BiDi Special Format

Managed (J2C) connection factory properties

The following managed (J2C) connection properties can be set to control bidirectional transformation.

- Output directory BiDi Format
- Output directory BiDi Special Format
- Skip BiDi Transformation for Output Directory
- Skip BiDi Transformation for Staging Directory
- Staging directory BiDi Format
- Staging directory BiDi Special Format

Activation specification properties

The following activation specification properties can be set to control bidirectional transformation.

- Event Directory BiDi Format
- Skip BiDi Transformation for Event Directory
- Event Directory BiDi Special Format
- Archive Directory BiDi Format
- Skip BiDi Transformation for Archive Directory
- Archive Directory BiDi Special Format
- Failed Archive Extension BiDi Format
- Skip BiDi Transformation for Failed Archive Extension
- Original Archive Extension BiDi Format
- Skip BiDi Transformation for Original Archive Extension
- Success Archive Extension BiDi Format
- Skip BiDi Transformation for Success Archive Extension
- Event File Mask BiDi Format
- Skip BiDi Transformation for Event File Mask
- Event File Mask BiDi Special Format
- Split Criteria BiDi Format
- Skip BiDi Transformation for Split Criteria
- Skip BiDi Transformation
- EIS BiDi Format
- EIS BiDi Special Format
- Event Persistence BiDi Format
- Skip BiDi Transformation for Event Persistence
- Data Source Name BiDi Special Format

- Skip BiDi Transformation for Data Source Name

Messages

The messages issued by IBM WebSphere Adapters are documented in the WebSphere Adapters, version 6.0.2 information center.

You can view the adapter messages at the following link: WebSphere Adapters messages..

Related product information

The following links, information centers, Redbooks, and Web pages contain related information for the IBM WebSphere Adapter for Flat Files.

Additional information you might need

Table 14. WebSphere Adapters information you might need

Information	How to find it
How to edit business objects using the Business Object Editor	In the IBM WebSphere Business Process Management information center, which includes documentation for WebSphere Integration Developer, search for the topic, "Editing Business Objects."
How to uninstall a deployed adapter	On the WebSphere Application Server library page, open the information center for your version of WebSphere Application Server and search for the topic, "Uninstalling applications."

Information for related products

- WebSphere Adapters, Version 6.0
- WebSphere Business Integration Adapters
- WebSphere Integration Developer
- WebSphere Process Server
- WebSphere Enterprise Service Bus
- WebSphere Application Server

Redbooks

- WebSphere Adapter Development Redbook
- WebSphere Redbooks domain

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

- WebSphere Adapters product support
- WebSphere Adapters technotes - in the **Additional search terms** field, specify the name of the adapter and click **Go**.

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, and 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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

IBM and related trademarks: <http://www.ibm.com/legal/copytrade.shtml>

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/>).

Index

A

- accessibility 5
 - administrative console 6
 - enterprise service discovery wizard 6
 - IBM Accessibility Center 6
 - installation 5
 - keyboard 6
 - shortcut keys 6
- activation specification 124
- activation specification properties
 - Archive Directory 124
 - ArchivingProcessed 124
 - AssuredOnceDelivery 124
 - CreateTable 124
 - DatabasePassword 124
 - DatabaseSchemaName 124
 - Default Object Name 124
- adapter
 - messages 131
- adapter configuration properties 121
- adapter project
 - creating 31, 74, 93, 107, 112
 - exporting 55
- adapter tasks 24
- adapter technotes 131
- adapters
 - uninstalling deployed 131
- administrative console
 - changing adapter configuration properties 57
 - setting activation specification properties 59
 - setting adapter configuration properties 57
 - setting managed connection properties 59
 - setting resource adapter properties 58
- append 8

B

- backward compatibility 28
- bidirectional transformation 19
- Business Object Editor information 131
- business objects 14
 - attribute properties 16
 - editing 131
 - naming convention 14
 - structure 14
 - supported operations 16

C

- clustered environment 23
- Common Event Infrastructure (CEI)
 - enabling tracing 61
- configuration overview 25
- Configuring the selected objects: Inbound 47, 102, 110
- Configuring the selected objects: Outbound 39, 78
- connection properties 119
- create 8

D

- data source
 - configuring 96
- data transformation framework 9
- DatabaseUsername 124
- DataSourceJNDIName 124
- debugging
 - controlling tracing details with CEI 61
 - enabling tracing with CEI 61
 - self-help resources 69
 - XAResourceNotAvailableException exception 68
- debugging tools
 - configuring 61
 - IBM Support Assistant 65
- delete 8
- deployment overview 26
- developerWorks, adapter-related resources 131

E

- enableHASupport 23
- enterprise service discovery 26, 119, 121
 - connection properties 36, 43
 - generating inbound artifacts 48
 - generating outbound artifacts 40
 - overview 17
 - selecting inbound business objects 46
 - selecting inbound services 46
 - selecting outbound business objects 38
 - selecting outbound services 38
 - setting connection properties 35, 43
- enterprise service discovery wizard
 - accessibility 6
- Event archival values 11
- Event Content Type 124
- Event Directory 124
- event store
 - overview 9
 - structure 10
- EventFileMask 124
- EventTableName 124
- EventTypeFilter 124
- exceptions
 - XAResourceNotAvailableException 68
- exists 8

F

- Failed Archive Extension 124
- File Content Encoding 124
- File Pass By Reference 124
- file splitting
 - delimiter based 12
 - size based 12
- files
 - RAR files for adapters 64
 - SystemOut.log log file 64
 - trace.log trace file 64
- FilterFutureEvents 124

G

globalization 19

H

high availability 23

I

IBM Support Assistant (ISA)

- installing 65
- overview 65
- plug-ins 65
- upgrading 65

IBM WebSphere Adapter for Flat Files

- uninstalling 30

IBM WebSphere Adapter Toolkit 131

inbound

- data transformation framework 13
- data transformation processing tutorial 93
- processing 9

Include End BO Delimiter 124

information centers, related 131

installation overview 25

installation pre-requisites

- hardware requirements 27
- software requirements 27
- supported platforms 27

Internet Protocol Version 6.0 (IPv6) 6

IPv6 6

K

keyboard 6

L

list 8

Log Analyzer, setting file format for 62

log files

- changing file name 64
- disabling 62
- enabling 62
- level of detail 62
- locating 64

logging

- configuring properties 62

M

managed connection factory properties 123

managed connection properties

- output directory 123
- OutputFileName 123
- staging directory 123

messages 131

migrating 28

migration

- performing 29

O

Original Archive Extension 124

outbound

- data transformation framework 9
- data transformation processing tutorial 74
- processing 8
- supported operations 8
- testing append 89
- testing create 81
- testing delete 92
- testing exists 83
- testing list 85
- testing overwrite 90
- testing retrieve 87

outbound operations

- append 8
- create 8
- delete 8
- exists 8
- list 8
- overwrite 8
- retrieve 8

overwrite 8

P

package files for adapters 63

passing parameters 8

PollPeriod 124

PollQuantity 124

problem determination

- configuring 61
- controlling tracing details with CEI 61
- enabling tracing with CEI 61
- self help resources 69
- XAResourceNotAvailableException exception 68

problem determination tools

- IBM Support Assistant 65

product information, related 131

product plug-ins

- for IBM Support Assistant 65

properties

- activation specification 124
- bidirectional transformation 130
- DatabaseUsername 124
- DataSourceJNDIName 124
- Event Content Type 124
- Event Directory 124
- EventFileMask 124
- EventTableName 124
- EventTypeFilter 124
- Failed Archive Extension 124
- File Content Encoding 124
- File Pass By Reference 124
- FilterFutureEvents 124
- Include End BO Delimiter 124
- managed connection factory 123
- Original Archive Extension 124
- PollPeriod 124
- PollQuantity 124
- resource adapter 121
- Retry Interval 124
- Retry Limit 124
- Sort Event Files 124
- Split Criteria 124
- Splitting Function Class Name 124
- Stop polling on error 124
- Success Archive Extension 124

Q

- quick start tutorials
 - creating an adapter project 74, 93, 107, 112
 - introduction 73
 - troubleshooting 117
 - tutorial 1: outbound processing with data transformation 74
 - tutorial 2: inbound processing with data transformation 93

R

- RAR files for adapters 64
- Redbooks, adapter-related 131
- reference bindings
 - generating 51, 105, 113
- replicated adapter instances 23
- resource adapter properties 121
 - Adapter ID 121
 - enableHASupport 121
 - Log file max size 121
 - Log file name 121
 - Log files 121
 - Trace file max size 121
 - Trace file name 121
 - Trace files 121
- retrieve 8
- Retry Interval 124
- Retry Limit 124

S

- security 23
- self help resources 69
- severity criteria for software problems 70
- shortcut keys 6
- Sort Event Files 124
- Split Criteria 124
- Splitting Function Class Name 124
- standards compliance 5
- Stop polling on error 124
- Success Archive Extension 124
- support
 - contacting 69
 - IBM Support Assistant 65
 - overview 68
 - self help resources 69
- support, technical 131

T

- task roadmap 24
- technical overview 7
- technical support 131
- technotes, adapter 131
- tools
 - for troubleshooting 61
- trace files
 - changing file name 64
 - disabling 62
 - enabling 62
 - level of detail 62
 - locating 64
- tracing
 - configuring properties with administrative console 62

- tracing (*continued*)
 - configuring properties with CEI 61
- troubleshooting 131
 - controlling tracing details with CEI 61
 - enabling tracing with CEI 61
 - overview 68
 - self help resources 69
 - XAResourceNotAvailableException exception 68
- troubleshooting tools
 - configuring 61
 - IBM Support Assistant 65
- tutorial 1
 - configuring outbound processing 75
 - generating artifacts 78
 - generating outbound business objects 35
 - selecting business objects and services 77
 - setting connection properties 75
- tutorial 2
 - generating artifacts 103
 - generating business objects 43
 - selecting business objects 101
 - setting connection properties 100
- Tutorial 2
 - configuring inbound processing 100, 109

U

- uninstalling deployed adapters 131
- uninstalling the adapter 30

W

- WebSphere Adapter for Flat Files 121, 123
 - accessibility 5
 - administering 67
 - backward compatibility 28
 - business objects 14
 - changing adapter configuration properties 57
 - creating the adapter project 31
 - inbound processing 9
 - installation pre-requisites 27
 - installing 27
 - introduction 5
 - migrating 28
 - outbound processing 8
 - performing the migration 29
 - planning adapter implementation 23
 - reference 119
 - security 23
 - setting adapter configuration properties 57
 - standards compliance 5
 - starting 67
 - stopping 67
 - technical overview 7
- WebSphere Adapters, version 6.0 information 131
- WebSphere Application Server information 131
- WebSphere Business Integration Adapters information 131
 - information
 - WebSphere Adapters, version 6.0 131
 - WebSphere Adapters, version 6.0.2 131
 - WebSphere Application Server 131
 - WebSphere Business Integration Adapters 131
 - WebSphere Enterprise Service Bus 131
 - WebSphere Integration Developer 131
 - WebSphere Process Server 131
- Redbooks 131

WebSphere Business Integration Adapters information	
(<i>continued</i>)	
WebSphere Adapters, version 6.0 information	131
WebSphere Adapters, version 6.0.2 information	131
WebSphere Application Server information	131
WebSphere Enterprise Service Bus information	131
WebSphere Integration Developer	
configuring the data source	96
creating an adapter project	74, 93, 107, 112
creating the adapter project	31
WebSphere Integration Developer information	131
WebSphere Process Server information	131



Printed in USA