WebSphere® Adapters

Version 6.0.2

Adapter for FTP User Guide

Note
Before using this information, be sure to read the general information in “Notices” on page 141.

22December2006

This edition applies to version 6, release 0, modification 2 of WebSphere Adapter for FTP (product number 5724-N42) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

Contents

Chapter 1. About this information

This documentation is for integration developers who implement, configure, and deploy WebSphere® Adapter for FTP. 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 FTP in a solution that requires data exchange between enterprise information systems (EIS) and J2EE applications using File Transfer Protocol (FTP). 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.

Chapter 2. Release notes

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

Release notes for WebSphere Adapter for FTP, version 6.0.2

Chapter 3. Introduction to WebSphere Adapters

IBM® WebSphere Adapters make it possible for Java™ 2 Platform, Enterprise Edition (J2EE) components, such as new e-business applications, to communicate with resources on an enterprise information system (EIS). An EIS is the information infrastructure for an enterprise (for example, an enterprise resource planning [ERP] system or an FTP server).

A WebSphere adapter acts as an intermediary between the J2EE component and the FTP server, so that the J2EE component does not need to understand the low-level API or data structures of the FTP server.

WebSphere Adapters can be one of two types: application or technology.

- Application adapters connect to existing packaged applications (such as SAP Software, Siebel, PeopleSoft Enterprise, and JD Edwards EnterpriseOne) so that you can make use of data and services specific to the applications.
- Technology adapters provide connectivity to data through such technologies and protocols as relational databases, flat files, e-mail messages, and FTP.

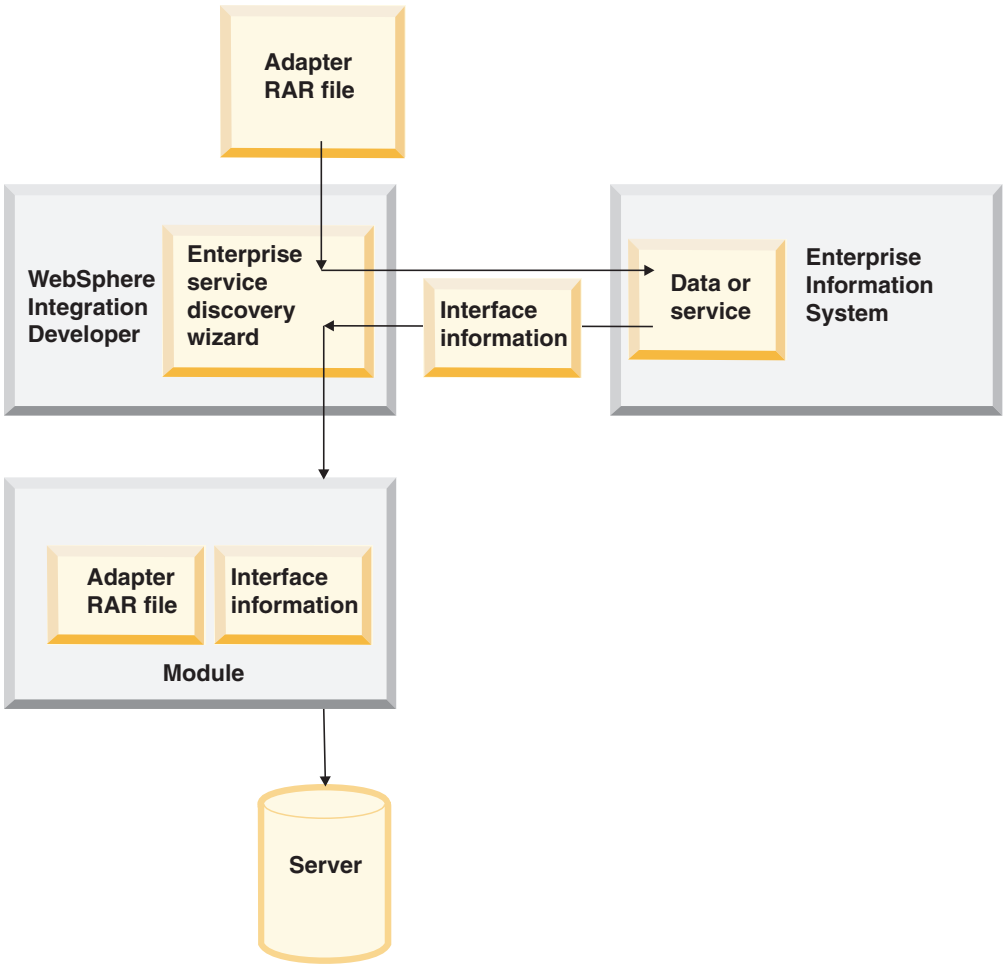
As part of the WebSphere family of products, WebSphere Adapters work with WebSphere Integration Developer and either WebSphere Process Server or WebSphere Enterprise Service Bus.

- WebSphere Integration Developer is the tooling environment for the WebSphere adapters.

  You use WebSphere Integration Developer to assemble a module that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus. From within WebSphere Integration Developer, you import the adapter (which is packaged as a resource adapter [RAR] file). You also have the pre-generated schema definitions for the business data in a folder. The enterprise service discovery wizard of WebSphere Integration Developer looks into these schema definitions and creates the interface information needed to gain access to the data and services. Finally, WebSphere Integration Developer generates a module that includes the adapter and the interface information.
- WebSphere Process Server or WebSphere Enterprise Service Bus is the runtime environment for the WebSphere adapters.

  You deploy the module generated by WebSphere Integration Developer to one of the servers.

The generation and deployment of the module is illustrated in the following figure.



How a module is generated and deployed

Chapter 4. Introduction to the WebSphere Adapter for FTP

IBM WebSphere Adapter for FTP connects Java 2 Platform, Enterprise Edition (J2EE) components running on WebSphere Process Server or WebSphere Enterprise Service Bus with remote file systems through an FTP server. The adapter connects to the FTP server to retrieve and write to the files. The adapter provides a means for the J2EE component and the remote file system to interact. For example, the J2EE application can be configured to use the adapter to update a customer record in the remote file system.

Hardware and software requirements

Before installing Adapter for FTP, 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 FTP, 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).

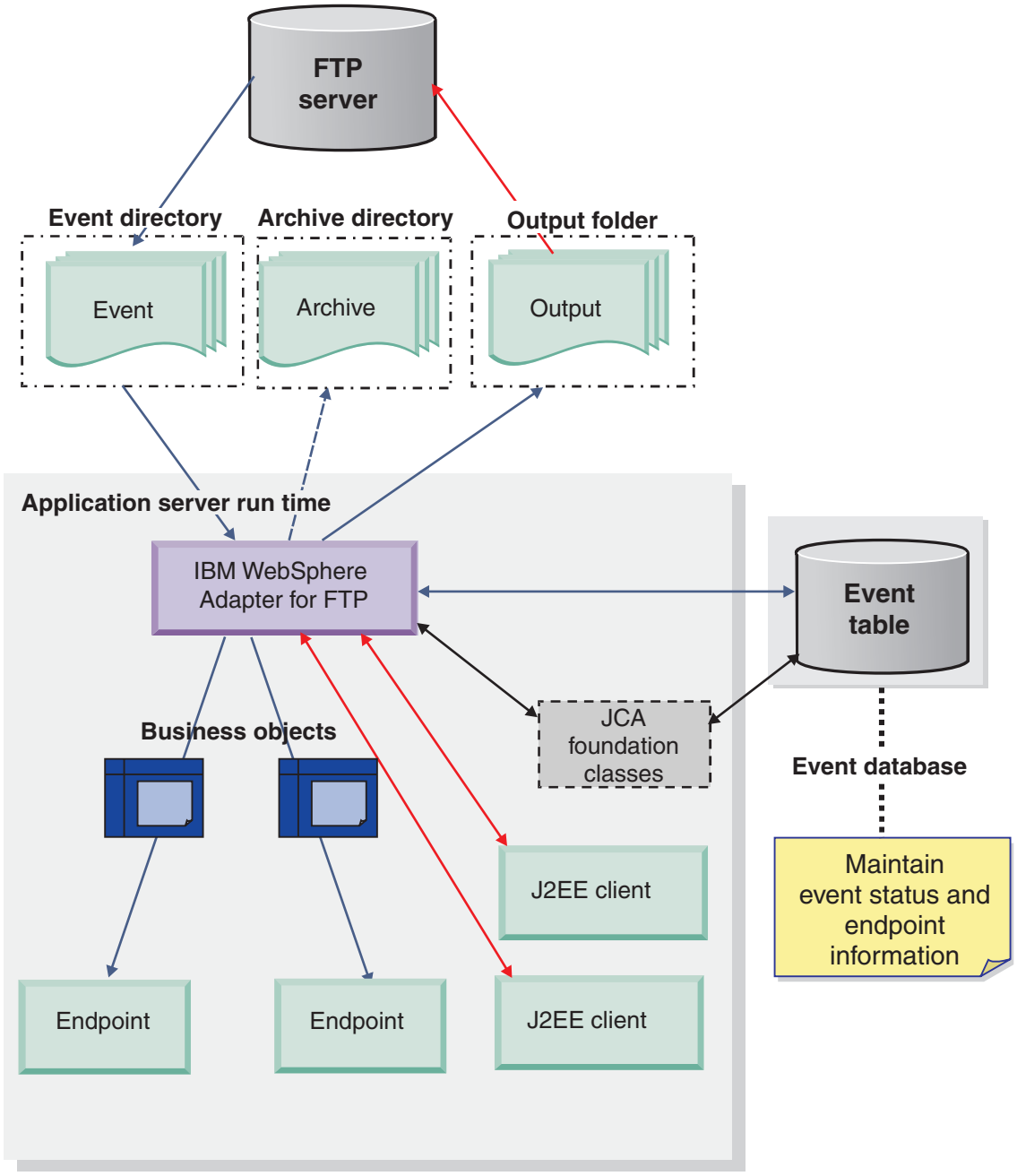
For more information about this compatibility in WebSphere Application Server, see IPv6 support in the WebSphere Application Server information center.

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

Technical overview of the Adapter for FTP

The FTP adapter supports the exchange of business data between remote file systems and J2EE applications by connecting to the FTP server for retrieving and writing to the files.

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



Adapter architecture diagram

Outbound processing

WebSphere Adapter for FTP 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. The adapter processes the request and returns a business object that represents the result of the operation to the J2EE application. The response to the J2EE application can be either a business object with the resulting data or `null`.

Supported outbound operations

WebSphere Adapter for FTP supports the operations described in the following table for outbound processing.

Operation	Result
Create	A file with the specified name is created in the specified directory of the FTP server. The content of the file arrives as either part of the request from the client or it can be retrieved from the local file system where the adapter is running. This content retrieval is configurable. If the file to be created does not exist, the file is created and null is returned to the J2EE application. The file is created in a staging directory (if present) and from there it is moved to the specified directory. If a staging directory is not present, the file is created in the specified directory. If the file to be created already exists, an FTPFileCreateException exception is sent, the file is not created, and the existing file is not overwritten.
Append	The file with the specified name in the specified directory of the FTP server is appended with the content sent across in the request. If the file exists, the content is appended and `null` is returned to the J2EE application. The file to be appended is copied from the specified directory to the staging directory (if present) and the content is appended to that file in the staging directory. The file is then moved back to the original specified directory. If the staging directory is not present, the content is directly appended to the file in the specified directory. If the file to be appended does not exist, an FTPFileAppendException exception is sent to the calling component.
Delete	The file in the specified directory is deleted on the FTP server. After deleting the file, `null` is returned to the J2EE application. If the file does not exist, an FTPFileDeleteException exception is sent to the calling component.
Retrieve	The content of the file, or files, that is specified in the request is returned. If the file, or files, specified in the request exists, the content of the file is retrieved and sent as the response. The file content can either be sent back to the J2EE application as business object or it can be saved to the local file system where the adapter is running. This is configurable. The file names are input as a comma separated list in the fileName attribute. If the files do not exist, an FTPFileRetrieveException exception is sent to the calling component.
Overwrite	This operation overwrites the file in the directory with the content specified in the request. If the file to be overwritten exists, the content is overwritten and `null` is returned to the J2EE application. The file to be overwritten is copied from the specified directory to the staging directory (if used) and the content is overwritten for that file in the staging directory. The file is then moved back to the original specified directory. If the staging directory is not present, the content is overwritten on the file in the specified directory. If the file to be updated does not exist, an FTPFileOverwriteException exception is sent to the calling component.
Exists	If the file name in the request exists in the specified directory, the adapter returns the ExistsResponse business object, with the DoesFileExists attribute set to `true`. If the file name does not exist, or the directory does not exist, the adapter returns the ExistsResponse business object, with the DoesFileExists attribute set to `false`.
List	This operation returns all the file names and subdirectory names in the directory specified in the request. The file names and subdirectory names in the directory are retrieved and sent as a response in the ListResponse business object. If the specified directory does not exist, an FTPFileListException exception is sent to the calling component.
ServerToServer FileTransfer	Transfers the specified file from one FTP server directory to another FTP server directory. If the request does not contain all of the information about the two servers, the adapter sends an FTPFileServerToServer FileTransferException exception.

Operation	Result
ExecuteFTPScript	This operation runs the commands present in an FTP script file in WebSphere Process Server. The operation runs only those commands that are supported by the FTP server and ignores the rest. If the operation fails, it sends an FTPFileExecuteFTPScriptException exception. The script file must not contain connection related commands such as `open` or `connect` because the adapter uses an already established connection to run the commands. If the commands in the script file need to be run in a particular directory on the FTP server, then the script file must contain the first command to change to that directory.

Outbound configuration options

WebSphere Adapter for FTP supports the following configuration options for outbound processing:

- Outbound operations support both active and passive data connection modes. The default mode is `active`.
- Files can be transferred in either ASCII or binary mode. The default is `binary`.
- You can enable auditing of the commands that are being run during an outbound operation. This information is written to the trace file when the trace level is set to FINEST.
- Socks proxy settings can be configured if the requests are being passed through an FTP proxy located between the adapter workstation and the FTP server. The proxy settings are configured at the application level and are common for all the ManagedConnectionFactory's and ActivationSpec's used during adapter configuration.
- If file content is sent from a J2EE application during an outbound create operation, an option is provided to save the file to the adapter workstation before performing the operation.
- For multiple files to be retrieved during outbound operations, the fileName attribute must contain a comma separated list of file names.
- ExecuteFTPScript can run before and after outbound operations The script file must not contain connection related commands such as `open` as we use an already established connection to run the commands.
- The port number used by the FTP server can be configured in the FTP URL. For Example: ftp://localhost:portNumber
- The passive data connection mode can be used to connect the adapter to an FTP server through a firewall.
- Default passwords are stored in plain text in the import and export files. If an Authentication Alias is configured, the actual password is taken from the Authentication Alias of WebSphere Process Server for both inbound and outbound configurations.
- If using an FTP server when the output of a long list command (ls -l) is different from the default listings supported by Apache commons net 1.4.1 for a different operating system, users can implement their own custom parser to parse the ls -l output.
- The adapter supports Secure FTP using SSL. It uses passive data connection mode during secure operations.
- If the FTP connection times out, the adapter detects and reestablishes the lost connection. If unable to establish the connection, the adapter retries for a fixed number of times. The interrupted transfers are not continued. They are lost and the file transfer must be restarted.

- During outbound create operations, files are created on the staging directory (if present). They are then moved to the actual required directory. This is helpful when a third application starts reading the file before its fully written. The staging directory is also used for Append and Overwrite operations where the specified file is copied to staging directory (if present) first, then appended or overwritten with content and moved back to the actual required directory. If a staging directory is not present, the operation is run in the actual required directory.

Outbound processing results

This section provides a detailed description of the business objects that are returned to the J2EE application upon completion of an outbound processing operation.

After outbound processing is completed, the adapter returns the resulting output to the calling J2EE Client. The following operations return `null`:

- Create
- Append
- Overwrite
- Delete
- ServerToServerFileTransfer
- ExecuteFTPScript

Exists operation

The following operations return business objects:

The Exists operation returns a business object containing information about the status of the operation. The structure of the business object is:

ExistsResponseBG->ExistsResponse

The status of the Exists operation is provided in the Boolean type element DoesFileExists within the ExistsResponse business object.

List operation

The List operation returns a business object containing information about the status of the operation. The structure of the business object is:

ListResponseBG->ListResponse

ListResponse will have one element, ListOfFileNames of type String[] which contains the files names present in the directory.

Retrieve operation

The Retrieve operation returns a business object containing information about the status of the operation. The structure of the business object is:

RetrieveResponseWrapperBG->RetrieveResponseWrapper->FileContent

The RetrieveResponseWrapper business object contains the Content element of type anyType[], which contains FileContent.

The FileContent business object contains the Filename element of type String and the Content element of type UnstructuredContent.

Elements under the RetrieveResponseWrapper business object

Element name	Type
Content	anyType[]
SavedFileToLocalDirectory	Boolean

If SavedFileToLocalDirectory is set to `true`, all files are saved to a local directory and the business object's Content element is set to `null`.

Elements under the FileContent business object

Element name	Type
Filename	String
Content	UnstructuredContent

Note: Data transformation will not happen while the output of the retrieve operation is being returned to the J2EE Client. In other words, the file content is not transformed into Customer or Employee business objects. The content will be in the form of bytes[] in the data objects.

Data Transformation Framework

During outbound processing the adapter uses data transformation to convert the data contained in WebSphere Adapter business objects to 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 Data Transformation Framework enables users to bridge this communication gap.

The following steps describe how data transformation occurs during outbound processing:

1. A wrapper business object (CustomerWrapper or FTPFile) contained in the wrapper business graph (CustomerWrapperBG or FTPFileBG) is populated with protocol specific information in the J2EE component. The actual business object (for example, Customer) is also set in the wrapper business object. This wrapper business object and the outbound operation name are sent as input through an SCA call.
2. Based on the data binding configured while running the enterprise service discovery wizard, the correct data binding is called and this data binding gets the WrapperBG business object.
3. The adapter checks the annotation of the wrapper business object and the Customer business object. Based on the content type set in the annotation of the Customer business object, the correct content-specific data binding is invoked. The mapping between the content type and the content-specific data binding to be called is obtained from the annotation of the wrapper business object. If the content type is invalid or the content-specific data binding name is invalid, then an appropriate error is generated by the Data Transformation Framework. The following example shows the annotation for both CustomerWrapper and Customer:

```
<complexType name="CustomerWrapper">
<annotation>
<appinfo source="http://www.ibm.com/xmlns/prod/websphere/
j2ca/datatransformation/databindingmapping"><dtm:DataBindingMapping
xsi:type="dtm:DataBindingMapping" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xmlns:dtm="http://www.ibm.com/xmlns/prod/
websphere/j2ca/datatransformation/databindingmapping">
<DataBindingConfig xsi:type="dtm:XMLBOSerializerDataBindingMetadata">
<ContentType>text/xml</ContentType>
<DataBindingClass>com.ibm.j2ca.extension.emd.runtime.WBIXMLDataBinding
</DataBindingClass>
</DataBindingConfig>
</dtm:DataBindingMapping>
</appinfo>
</annotation>
...
<complexType name="Customer">
<annotation>
<appinfo source="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
<dtm:DataTransformationMetadata xmlns:dtm="http://www.ibm.com/
xmlns/prod/websphere/j2ca/datatransformation/databindingmapping">
<ContentType>text/xml</ContentType>
<Charset></Charset>
<WrapperName>http://www.ibm.com/xmlns/prod/websphere/j2ca/
ftp/customerwrapperbg/CustomerWrapperBG</WrapperName>
</dtm:DataTransformationMetadata>
</appinfo>
</annotation>

<sequence maxOccurs="1" minOccurs="1">
<element maxOccurs="1" minOccurs="0" name="CustomerName" type="string"/>
<element maxOccurs="1" minOccurs="0" name="Address" type="string"/>
<element maxOccurs="1" minOccurs="0" name="City" type="string"/>
<element maxOccurs="1" minOccurs="0" name="State" type="string"/>
</sequence>
</complexType>
```

4. The request is sent to the adapter and the outbound operation is run. The adapter returns a business object that represents the result of the operation to the J2EE application component.

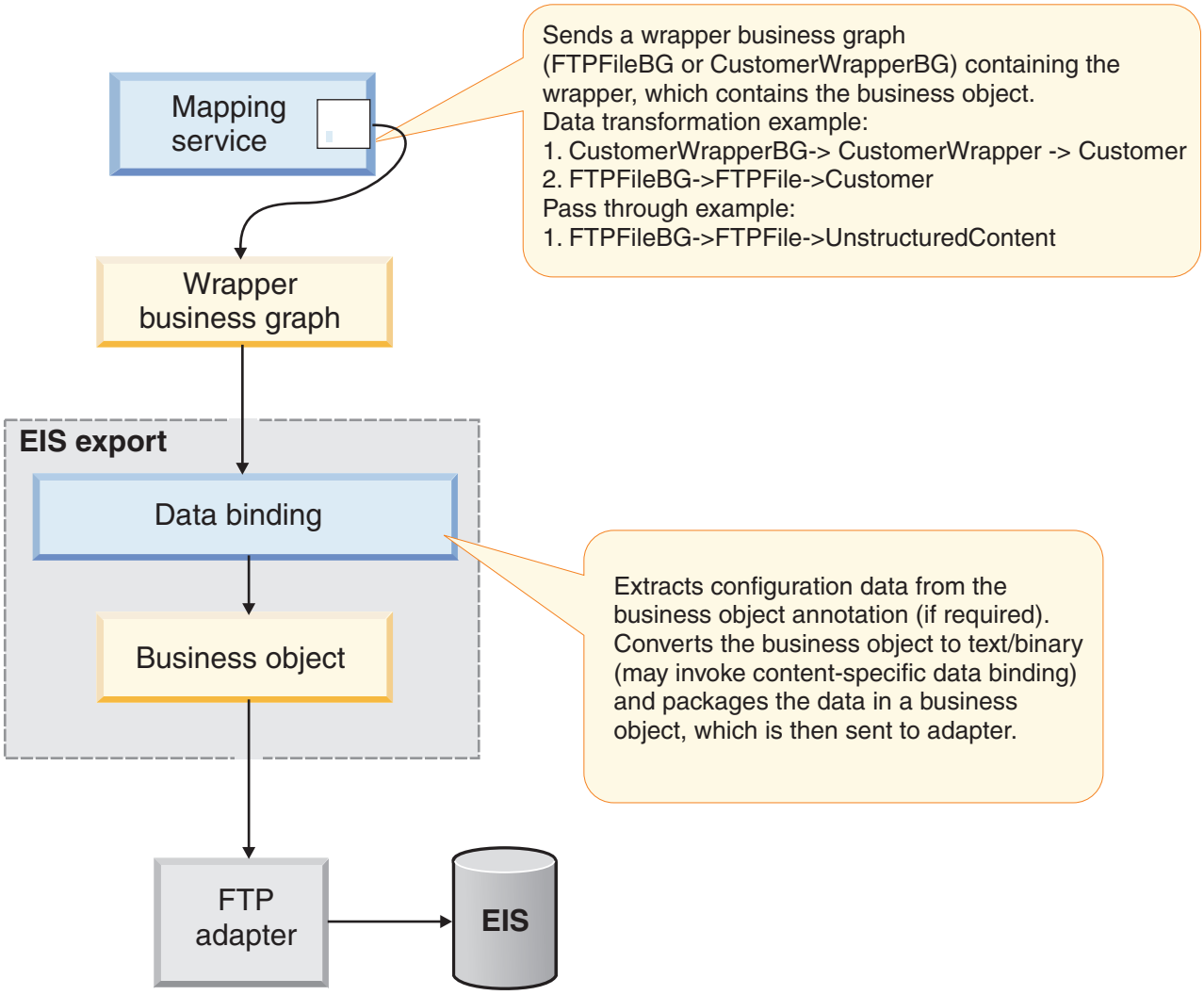
Pass through processing

Pass through processing occurs when data transformation will not happen. During pass through processing, the custom data binding that extends UnstructuredContentDataBinding does not call any other data binding.

The following steps describe how data pass through occurs during outbound processing:

1. The FTPFile wrapper business object contained in the FTPFileBG wrapper business graph is populated with protocol specific information in the J2EE component. The Content attribute of FTPFile business object is populated with an unstructured business object. This wrapper business object and the outbound operation name are sent as input through an SCA call.
2. Based on the data binding (FTPFileDataBinding) configured while running the enterprise service discovery wizard, the correct data binding is called and this data binding gets the FTPFileBG wrapper business object.
3. The FTPFileDataBinding recognizes the content as unstructured and performs pass through processing. It instantiates the FTPFileUnstructuredRecord, sets the actual content, and sets the protocol specific information.

4. This business object is sent to the adapter and the outbound operation is executed. The output of the outbound operation is a business object that is sent back to the WBIDataBindingImpl data binding and the business object is sent back to the J2EE client.



Outbound processing diagram

Parameter passing in the SCA framework

Parameters define the operation to be performed. You can use the service client to pass protocol-specific parameters, such as the directory path and file name.

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

- Set connection information in the ManagedConnectionFactory while running the enterprise service discovery wizard or set the information in the administrative console of WebSphere Process Server.
- Protocol-specific information such as directory name and file name and other properties required by specific outbound operations are set in the wrapper business object. These values are set in the business object in the custom data binding and sent to the adapter. These values are sent to the adapter for further processing.

- The file content is part of the input business object.

The adapter first searches for the UserName and Password values, if set, in the Authentication Alias. If not set there, it checks for the values in the ConnectionSpec. If the values are not set in the ConnectionSpec, it uses the values set in the Managed Connection Factory Properties window.

The mandatory property **Ftp Url** in the Managed Connection Factory Properties window is used to connect to the FTP server.

Inbound processing

Inbound event processing is an asynchronous operation. The adapter polls the FTP server, pulls events from the FTP server, converts the information into business objects, and sends the business objects to the configured endpoint.

The adapter polls files from the event directory of the FTP server at regular intervals based on the FTPPollFrequency property. When an file arrives in the event directory, the adapter reads the entire file and downloads the file to a local event directory on the adapter server. After the file is downloaded, the adapter either archives the file in the FTP sever in an archive directory given by the FTPArchiveDirectory property or deletes it based on your configuration. The event directory, archive directory, the poll interval, and the poll quantity (the number of files to poll in a single poll cycle) are all configurable parameters.

After the business objects are successfully posted to the endpoint, the events are either archived in an archive directory on the local file system or deleted, based on your configuration. The adapter must archive or delete the events or they will be polled again.

The adapter sends the business object to the endpoint via a function selector, which selects an operation to invoke on the component, and a data binding.

Inbound event processing consists of the following steps:

1. FTP server generates events in the form of files.
2. The FTP adapter polls the files from the event directory.
3. The files are fully downloaded to the adapter server.
4. The files are split based on the SplittingFunctionClassName and SplitCriteria properties.
   - If splitting needs to be done based on a delimiter, the class that performs this functionality and the split criteria are provided.
   - If splitting needs to be done based on file size, the class name that performs this functionality is provided.

   You can implement a custom class containing the splitting logic. The adapter provides a Java interface for the class. The details of the interface are shown below.

```
public interface SplittingFunctionalityInterface extends Iterator{
 public int getTotalBOs(String filename) throws SplittingException;
public void setBODetails(String filename, int currentPosition, int totalBOs,
 boolean includeEndBODelimiter) throws SplittingException;
 public void setSplitCriteria(String splitCriteria);
 public void setEncoding(String encoding);
 public void setLogUtils(LogUtils logUtils);
public boolean isSplitBySize()
}
```

   a. `public int getTotalBOs(String filename) throws SplittingException`

This method returns the total number of business object's present in the event file given by `filename`.

b. `public void setSplitCriteria(String splitCriteria)`

   This method takes the `splitCriteria`, which is based on the number of business object's in the event file. Each business object is returned during the `next()` call.

c. `public void setLogUtils(LogUtils logUtils)`

   This method is used to set the `LogUtils` object, which is the class that the user can use to write trace and log messages to the files.

d. `public void setEncoding(String encoding)`

   This method is used to set the encoding of the event file content. This encoding is used while reading the file content. This encoding is also used for the `SplitCriteria`.

e. `public void setBODetails(String filename, int currentPosition, int totalBOs, boolean includeEndBODelimiter) throws SplittingException`

   This method is used to set the current business object number so that whenever a `next()` call is made, the business object number set in the `currentPosition` is returned. It also takes an `includeEndBODelimiter` parameter, which when set to true, includes the `SplitCriteria` at the end of the business object content. This method must be called before every `next()` call so that the `next()` method returns the business object content for the business object set in this method.

f. The iterator has 3 methods: `hasNext()`, `next` and `remove()`, which also need to be implemented. The `next()` method returns the business object content for the business object position set in `setBODetails()`. If the business object position is not set, it fails. The `hasNext()` method indicates whether the business object position set in the `setBODetails()` exists or not. Before a `hasNext()` call, the `setBODetails()` method must be called. The `remove()` method is called for each of the business object entries being deleted from the EventPersistence table. Do not delete the event file in this method. Only clean up resources that are being used.

g. `public boolean isSplitBySize()`

   This method indicates whether the event file is parsed based on size or based on delimiter.

5. The adapter sends the business object to the endpoint through a function selector, where the configured data binding is invoked, converting the text record into a business object. The business object is sent to the endpoint.

Data Transformation Framework

During inbound processing, Data Transformation Framework converts the event data to a WebSphere Adapter business object. This conversion is necessary because the components in WebSphere Process Server consume only WebSphere Adapter business objects.

The following steps describe how data transformation occurs during inbound processing:

1. Each business object is selected from the event file based on the SplitCriteria property, which contains the delimiter that is used to separate the business objects in an event file.
2. The DefaultObjectName and EventContentType properties from the ActivationSpec are set to the corresponding objectName and contentType values in the business object.

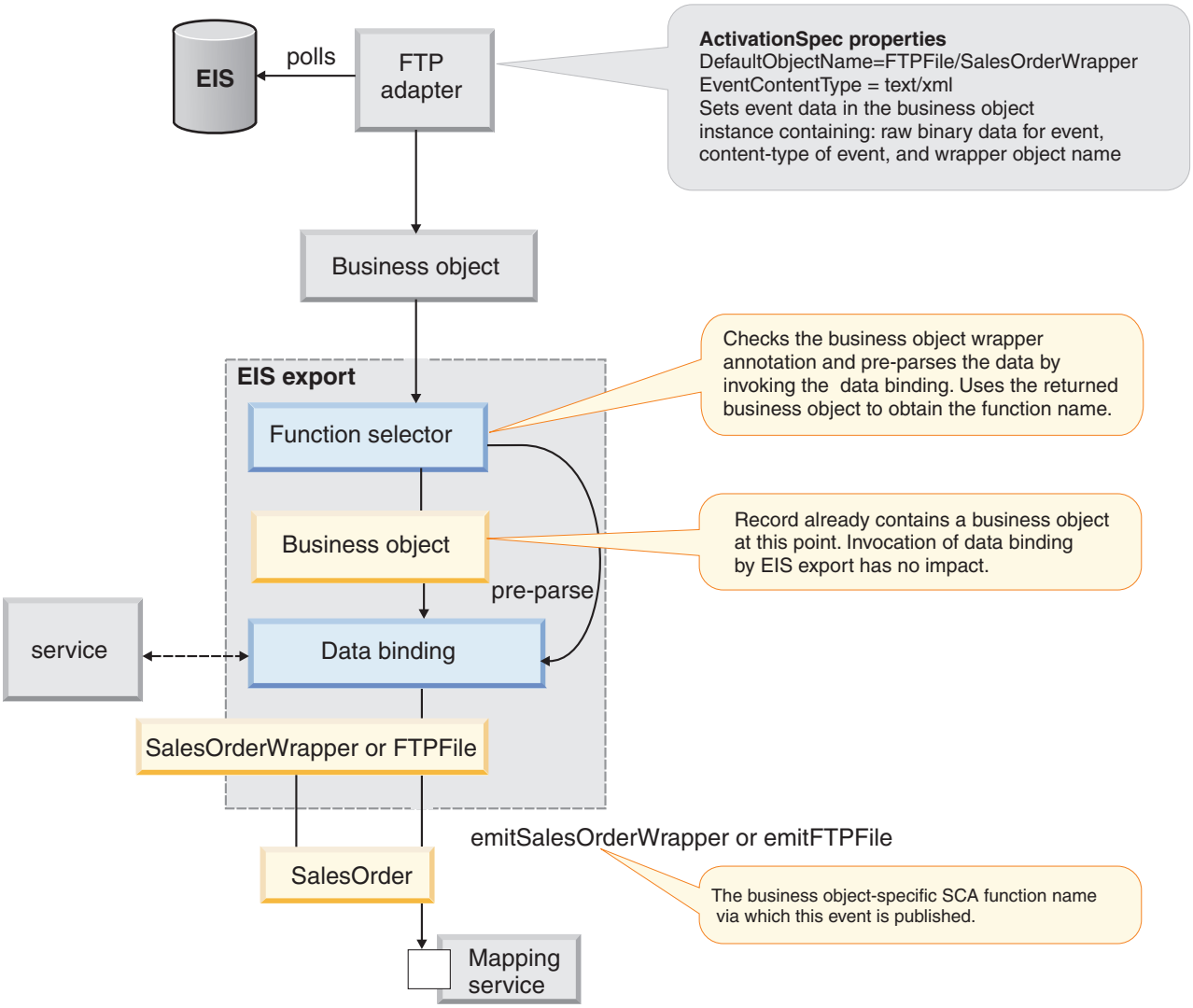
The business object name can be the name of a generic wrapper business graph, such as FTPFileBG or the name of the content-specific wrapper business graph, such as SalesOrderWrapperBG. The EventContentType property must be set to a valid value.

3. Protocol-specific information such as the event file name and the directory name are set in the business object.
4. The business object is sent to the function selector, which invokes the specified data binding.
5. The content-specific business object is set on the wrapper business object. Protocol-specific information, such as the directory name and the file name, is set in the wrapper. The wrapper is set in a business graph and sent to the endpoint. An example of the annotation for both the SalesorderWrapper and SalesOrder business objects is shown here.

```
<complexType name=" SalesorderWrapper ">
<annotation>
<appinfo source="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
<dtm:DataBindingMapping xsi:type="dtm:DataBindingMapping"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dtm="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
<DataBindingConfig xsi:type="dtm:XMLBOSerializerDataBindingMetadata">
<ContentType>text/xml</ContentType>
<DataBindingClass>com.ibm.j2ca.extension.emd.runtime.
WBIXMLDataBinding</DataBindingClass>
</DataBindingConfig>
</dtm:DataBindingMapping>
</appinfo>
</annotation>
 ...
<complexType name="FTPFile">
<annotation>
<appinfo source="http://www.ibm.com/xmlns/prod/websphere/
j2ca/datatransformation/
databindingmapping">
<dtm:DataBindingMapping xsi:type="dtm:DataBindingMapping"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:dtm="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
<DataBindingConfig xsi:type="dtm:XMLBOSerializerDataBindingMetadata">
<ContentType>text/xml</ContentType>
<DataBindingClass>com.ibm.j2ca.extension.emd.runtime.WBIXMLDataBinding</
DataBindingClass>
</DataBindingConfig>
</dtm:DataBindingMapping>
</appinfo>
</annotation>
 ...


<complexType name=" SalesOrder ">
<annotation>
<appinfo source="http://www.ibm.com/xmlns/prod/websphere/j2ca/
datatransformation/databindingmapping">
<dtm:DataTransformationMetadata xmlns:dtm="http://www.ibm.com/
xmlns/prod/websphere/j2ca/datatransformation/databindingmapping">
<ContentType>text/xml</ContentType>
<Charset></Charset>
<WrapperName>http://www.ibm.com/xmlns/prod/websphere/j2ca/ftp/
salesorderwrapperbg/SalesOrderWrapperBG</WrapperName>
</dtm:DataTransformationMetadata>
</appinfo>
</annotation>
```

```
 <xs:complexContent>
  <xs:sequence minOccurs="1" maxOccurs="1">
   <xs:element name="Id" type="xs:integer"/>
   <xs:element name="Priority" type="xs:integer" default="3"/>
   <xs:element name="Item"
    xmlns:child="http://... "
    type="child:OrderLineItem" minOccurs="0"/>
  </xs:sequence>
 </xs:complexContent>
</xs:complexType>
```



Inbound data transformation processing diagram

Pass through processing

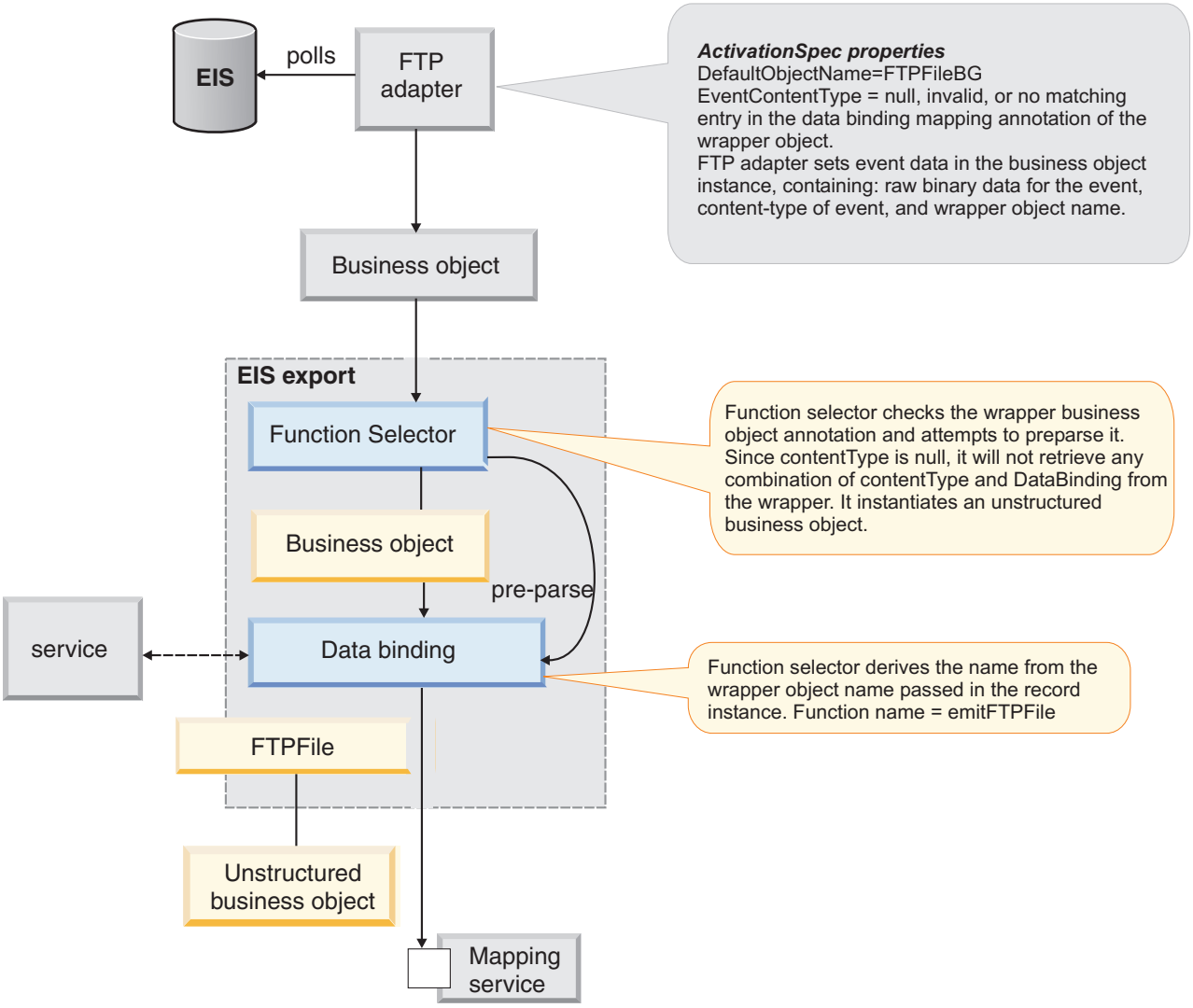
Pass through processing occurs when data transformation will not happen.

The following steps describe how data pass through occurs during inbound processing:

1. Each business object is retrieved from the event file based on the specified split criteria. The DefaultObjectName and EventContentType properties contained in the ActivationSpec are set to the corresponding record name and content type

contained in the unstructured business object. The default object name must be set to the name of generic wrapper business graph, for example, FTPFileBG.

2. Protocol-specific information like the event file name and the directory name are set in the unstructured business object. If the PassThrough has a chunk file or if it is FilePassByReference or default one is also indicated.
   - For FilePassByReference processing, the directory name corresponds to the local archive directory and the file name of the event file is appended with a timestamp.
   - For chunking processing, the directory name corresponds to the local archive directory and the filename represents the event file. The ChunkInfo property represents the chunk details.
   - For normal pass through processing, the directory name corresponds to the local archive directory and the filename represents the event file.
3. When the EventContentType is either set to `null`, or is invalid, or does not contain a matching entry in the data binding mapping annotation of the wrapper, the inbound scenario is processed as pass through. If the DefaultObjectName property is not, it is set to name of the generic wrapper business graph, for example, FTPFileBG.
4. The FTPFileUnstructuredRecord business object is sent to the function selector, which instantiates the wrapper and checks the annotation of the data binding mapping for the contentType and DataBinding property combinations. This allows pre-parsing to occur in the function selector based on the content type of the incoming FTPFileUnstructuredRecord business object. If no matching data binding is defined, for example, contentType is `null`, invalid, or does not contain an entry in the data binding mapping, or the invoked data binding could not resolve to a business object, no content-specific data binding is invoked and an unstructured content business object is instantiated.
5. The FTPFile wrapper business object is set with protocol-specific information and the UnstructuredContent business object is set in the FTPFile business object.
6. The FTPFile wrapper business object is set in the FTPFileBG business object and then sent to the endpoint.



Inbound passthrough processing diagram

Passing files by reference

The adapter also supports a PassThrough feature, where only the event file name is sent to the endpoint. The event file is appended with a time stamp and is available in the local archive directory. This feature is used when the endpoint applications have the capability to download the event files.

File splitting

The inbound event processing mode supports an optional file splitting feature, where the event file is split into several chunks and each chunk is posted to the endpoint separately. This reduces memory loading during event processing.

File splitting is performed based on either a delimiter or on a file size specified in the SplitCriteria property. The delimiter value is set in the SplitCriteria property and the class used to split the event file is set in the SplittingFunctionClassName property of the activation specification. By default the adapter provides the SplitBySize and SplitByDelimiter classes for the SplittingFunctionClassName property.

When chunking is enabled, each chunk of the file is posted to the endpoint separately. The number of business objects that are specified in the PollQuantity property is posted to the endpoint. For example, If the value for PollQuantity is 3, then:

Number of business objects polled is 3.

Number of business objects received by the endpoint is 3.

The adapter does not reassemble chunked data. It provides the information about the chunked data for an external application to merge the chunks. The chunking information is set in the chunkInfo property, which is contained in the wrapper business object. This information includes the chunk size, in bytes, and the event ID. An example event ID is:

```
AbsolutePathOfTheEventFileNameInLocalEventDirectory_/_yyyy_MM_dd_HH_mm_ss_SSS.
currentBONumber_/_totalBOs
```

Delimiter separating

Delimiters are used for separating event files. The delimiter is specified in the SplitCriteria property of the activation specification.

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 `(;;)` in front of it so that the content between the delimiters is skipped. For example, if the event file contains a business object in the following format and the delimiter is `##;;$$`, then:

  ```
  Name=Smith
  Company=IBM
  ##this is the content that will be skipped by the adapter$$
  ```

  The adapter will consider `##$$` the delimiter and skip `"this is the content that will be skipped by the adapter."`
- The delimiter takes any value and there are no restrictions. The following delimiters are valid:
  - `####;\n;\n`
  - `####;$$$$;\n;####`
  - `%%%%;$$$$$;#####`
  - `\n;\n;$$$$`
  - `####\;####;\n;$$$$$`
  - `\n;\n;\n`
  - `####;;$$$$`
- If the delimiter is located at the end of the file, the SplitCriteria property uses `END_OF_FILE` to determine the physical end of the file.

Event recovery

To use the recovery feature provided by WebSphere Process Server, you must set the AssuredOnceDelivery property in the ActivationSpec to `true`. If it is set to `false`, failed events can not be recovered. Duplicate events can be delivered if AssuredOnceDelivery is set to `false`. For improved performance, no event recovery, and duplicate events, set AssuredOnceDelivery to `false`.

Event store

Files created by the FTP server or any third party application in the event directory of the FTP server are called event files. Each business object record present in the event file is called an event. The adapter creates an entry in the database table, called an event store, to track each of these business object records until the event is posted to the endpoint.

If the database server goes down during an inbound operation, the adapter will send an exception. The database server and the adapter must be restarted to process the events.

Event table structure

Column Name	Type	Description
EVNTID	Varchar(255)	Each event requires a unique event ID for tracking purposes. The adapter uses this ID to track events during inbound operations.
EVNTSTAT	integer	The status of the event. The adapter uses the status to determine whether an event is new or in process. Event status values: **NEWEVENT (0)** The event is ready to be processed. **PROCESSED (1)** The adapter successfully processed and delivered the event. **FAILED (-1)** The adapter was unable to process this event due to one or more problems.
XID	Varchar(255)	Used by the adapter for assured event delivery and recovery.
EVNTDATA	Varchar(255)	Used by the adapter to mark the failed events as `ARCHIVED` to ensure that they are not processed again during adapter startup or recovery.

Event archive

Archived events are stored in the archive directory with a file extension that is specified in the FTPRenameExt property contained in the activation specification.

The following configurations apply to this option:

- When both the FTPArchiveDirectory and FTPRenameExt values are provided and FTPRenameExt is set to `processed`, the archived file is located in the specified archive directory in the following syntax:
  `filename_timestamp``.processed`

- When only the FTPArchiveDirectory value is provided, the archived file is located in the specified archive directory in the following syntax: *filename_timestamp*
- When both the FTPArchiveDirectory and FTPRenameExt values are not provided, the event file is deleted from the event directory of the FTP server after the file is successfully downloaded to the local event directory.
- When only the FTPRenameExt value is provided and is set to `processed`, the archived file will be located in the event directory of the FTP server in the following syntax: *filename_timestamp*`.processed`

Archiving on MVS™ platforms

Multiple Virtual Storage (MVS) operating systems do not support special characters such as `_`, in dataset or recordset names. On Windows and Unix platforms, use a time stamp in the original filename while archiving the file. This prevents duplicate file names in an archive folder, thereby preventing the overwriting of an existing file. Use the following format for MVS systems:

```
Event File: Test Archived

file: Test.TSyyyyMM.TSDDHHMM.TSSsSss
```

Where:

```
yyyy -- year

MM -- month

DD -- date

HH -- hour

MM -- minutes

Ss -- seconds

Sss -- milliseconds
```

The dataset or recordset separator is `.` (dot) on MVS platforms. The maximum number of `.` (dots) allowed in a dataset or recordset is 6. The dataset or recordset name must not exceed 8 characters per `.` (dot), and the total number of characters must not exceed 44 characters. An example of a file name in this format is:

```
FTPRenameExt: ARCHIVE

Archived File: TEST.TS200304.TS290535.TS42234.ARCHIVE
```

Inbound configuration options

WebSphere Adapter for FTP supports the following inbound processing configuration options:

- Event files can be archived at both the FTP server and in the local archive directory of the adapter workstation. If the FTPArchiveDirectory property has a valid directory value for the FTP server, the file is archived on the FTP server. If no valid directory value is provided, the event file is deleted after the download completes. In the adapter workstation, the event file is archived based on the

following property settings: LocalArchiveDirectory, OriginalArchiveExt, SuccessArchiveExt, and FailedArchiveExt.

- The adapter can retrieve files from the event directory based on the EventFileMask property setting. If EventFileMask is set to `*.*`, the adapter will retrieve all of the files in the event directory.
- Event files picked up from the event directory can be processed based on the order provided by the SortEventFiles property, which takes both `Timestamp` and `Filename` as values. If no value is given, the adapter does not sort the events, but processes them in the order they were received. Event ordering is only valid if the ActivationSpec DeliveryType property is set to `ORDERED`.
- Inbound operations support both active and passive connection modes. The default is active.
- The adapter supports Secure FTP using SSL. It uses passive data connection mode during secure operations.
- Passive data connection mode can be used to connect the adapter to an FTP server through a firewall.
- Files can be transferred in ASCII or binary mode. The default is binary.
- You can enable auditing of the commands that are being run during an inbound operation. This information is written to the trace file when the trace level is set to FINEST.
- You can configure Socks proxy settings if the requests are being passed through an FTP proxy located between the adapter workstation and the FTP server. The proxy settings are configured at the application level and are common for all the ManagedConnectionFactory's and ActivationSpec's used during adapter configuration.
- If you use an FTP server when the output of a long list command (`ls -l`) is different from the default listings supported by Apache commons net 1.4.1 for a different operating system, you can implement your own custom parser to parse the `ls -l` output.
- ExecuteFTPScript can run before and after outbound operations The script file must not contain connection related commands such as `open` as we use an already established connection to run the commands.
- For PassThrough, you have the option to split large event files based on configuration property settings. The file is downloaded to the local workstation from the FTP server, split into chunks and each chunk is posted to the endpoint.
- The port number used by the FTP server can be configured in the EventDirectory, for example: `ftp://localhost:portNumber/eventDirectory`.
- Default passwords are stored in plain text in the import and export files. If an Authentication Alias is configured, the actual password is taken from the Authentication Alias of WebSphere Process Server for both inbound and outbound configurations.
- During PassThrough operations, you can send only the file name to the endpoint using the FilePassByReference option. The event file (appended with the time stamp) is available in the local archive directory.
- The user can configure multiple ActivationSpecs so that adapter can poll multiple event directories in different FTP servers simultaneously and deliver to respective endpoints. For example, if the properties in two ActivationSpecs are same, the adapter polls the events once and delivers to two end points, one for each activation spec.

- If the FTP connection times out, the adapter detects and reestablishes the lost connection. If unable to establish the connection, the adapter retries for a fixed number of times. The interrupted transfers are not continued. They are lost and the file transfer must be restarted.

Business objects

Business objects carry the functional properties, data transformation information, and file content that the adapter needs to process requests and generate responses. The FTP adapter uses the same business object structure for both inbound and outbound processing. Depending on your business need, you can either 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 convention

Business object names must 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 FTP server.

For the FTP adapter, the business objects definitions (XSDs) are pregenerated and are available in a folder that we point to during enterprise service discovery.

The XSD's to be used during enterprise service discovery should be valid and the complexType name or element name should not contain any special characters. Any invalid XSD's will cause failures. Also, the XSD's should not contain any annotations as enterprise service discovery will add the required annotations.

The parent business object graph must 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.

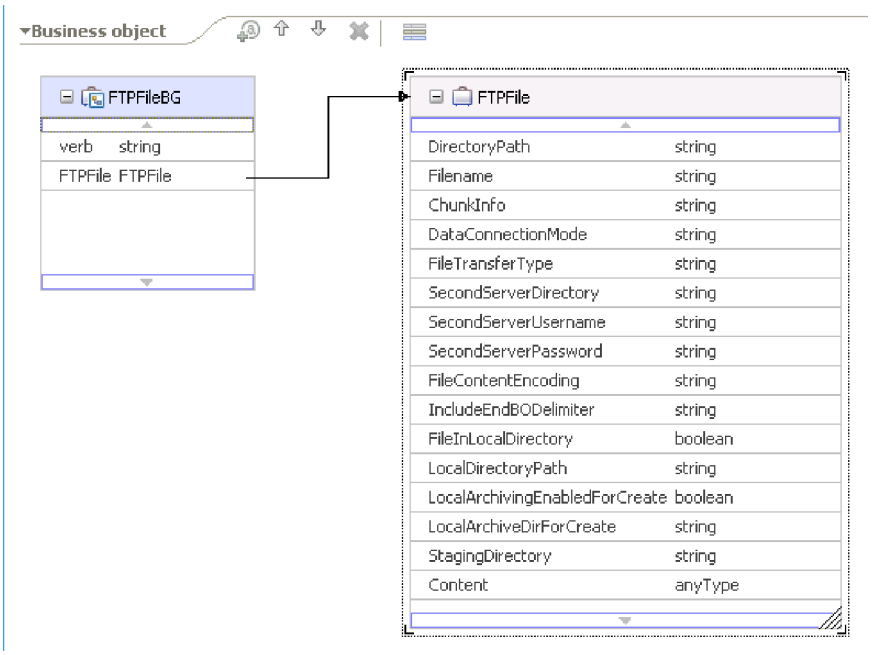
Business object structure

The FTP 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 FTP adapter defines and generates the following business objects during enterprise service discovery: FTPFile, FTPFileBG, Customer, CustomerWrapperBG, and CustomerWrapper.

The FTPFileBG, FTPFile and UnstructuredContent generic business object definitions are automatically generated. Depending on the custom complex types selected during enterprise service discovery, the corresponding business objects definitions will also be generated. For example, if we select Customer, the CustomerWrapperBG, CustomerWrapper, and Customer business objects will be generated.

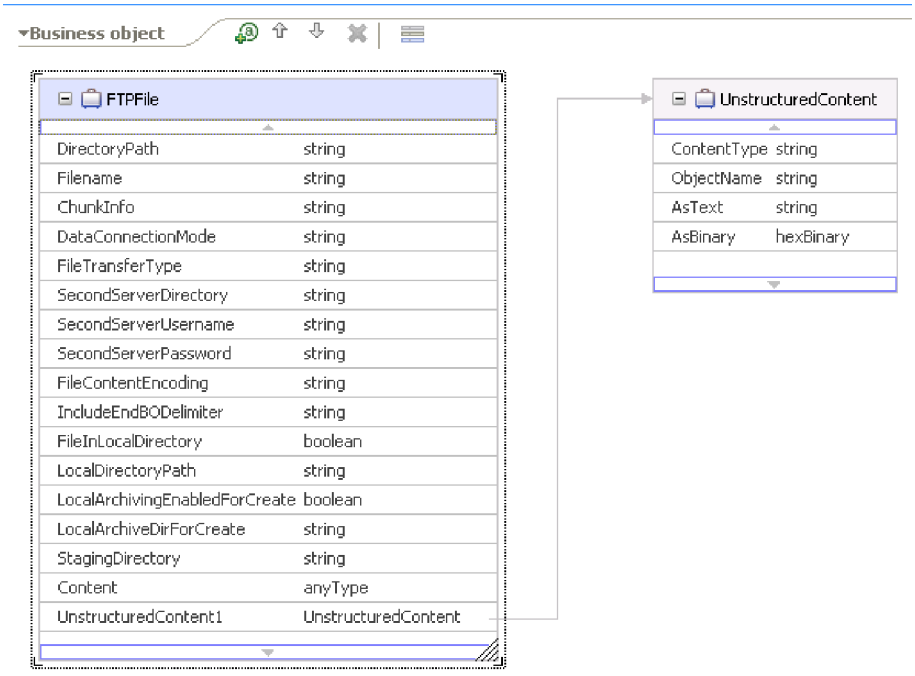
FTPFileBG

The FTPFileBG business object is a wrapper business object that contains the FTPFile business object as a child. The following graphic illustrates this relationship:



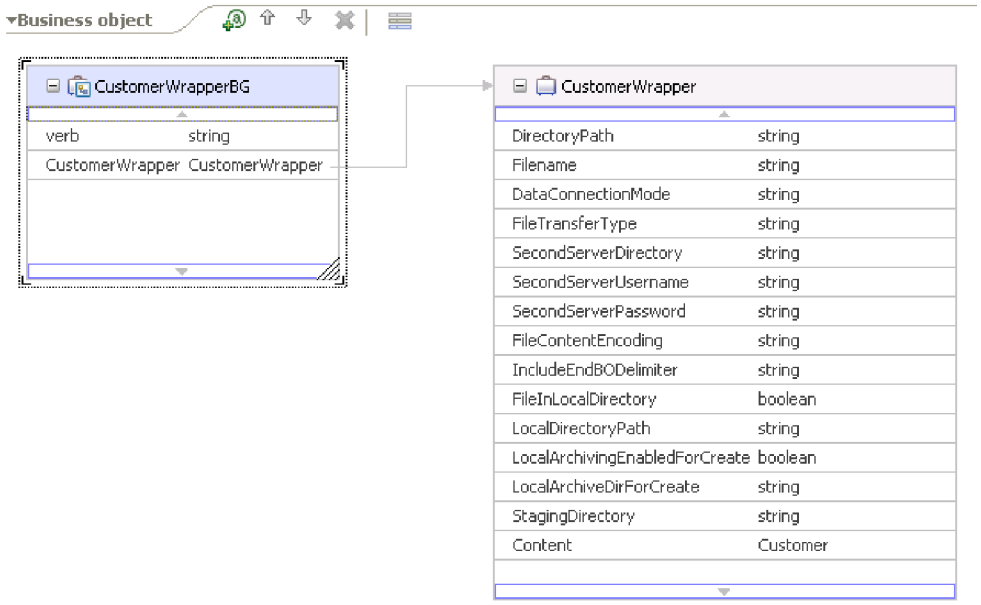
FTPFile

The FTPFile business object contains the UnstructuredContent business object as a child. The following graphic illustrates this relationship:



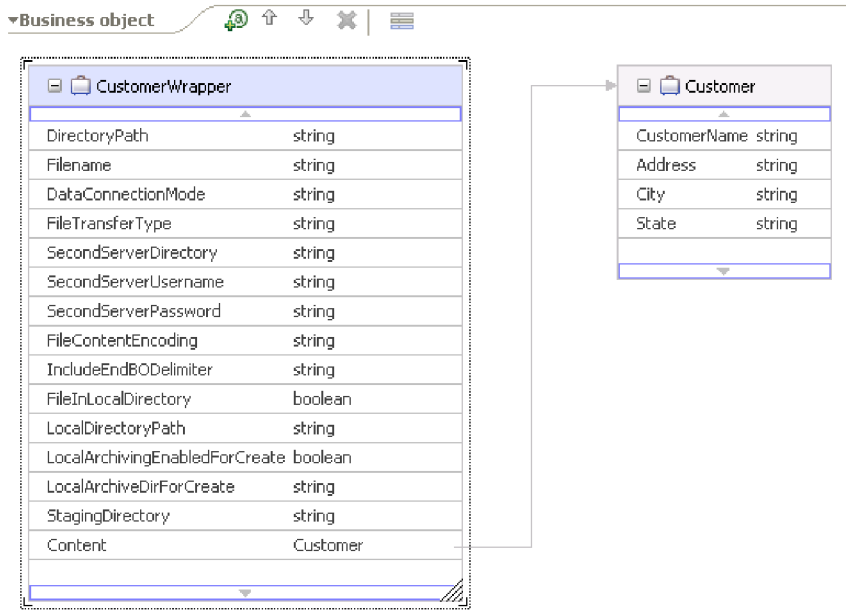
CustomerWrapperBG

The CustomerWrapperBG is a wrapper business object that contains the CustomerWrapper business object as a child. The following graphic illustrates this relationship:



CustomerWrapper

The CustomerWrapper business object is a wrapper business object that contains the Customer business object as a child. The following graphic illustrates this relationship:



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 describes these properties and how the adapter interprets them.

Property	Description
Cardinality	For simple attributes, 1 is used. For container attributes, depending on the method requirements, `n` is used.
Foreign Key	The adapter does not have any specific elements representing Foreign Keys.
Key	The adapter does not have any specific elements representing a Key.
Name	This property represents the unique name of the attribute, if it is a simple attribute, or the name of the business object, if it is a child business object.
Required	This property specifies whether an attribute must contain a value. The only required attribute is the DirectoryPath, which must be set for all outbound operations.
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 the name of another business object.

Business object operation support

The FTP adapter uses operations to specify the action to be performed on the FTP server during outbound processing. Refer to the following supported operations table for a list of the supported operations and the expected responses for each.

Supported operations for outbound processing

Operation	Result
Create	Creates a file with the specified file name in the specified directory with the content sent across in the request.
Append	Appends the content in the request to the end of the file.
Retrieve	Returns the content of the file specified in the request.
Delete	Deletes the file from the directory specified in the request.
Overwrite	Overwrites the file in the directory with the content specified in the request.
Exists	Returns a successful response if the file in the request exists in the specified directory.
List	Returns all the file names in the specified directory.
ServerToServerFileTransfer	Transfers the file from one FTP server to another FTP server.
ExecuteFTPScript	Runs an FTP script file in the specified directory.

Business object application-specific information

Business object application-specific information is located in the annotations of the wrapper business object and in the content-specific business object. The FTP adapter's enterprise metadata discovery (EMD) is responsible for generating the service descriptions and adding annotations for the selected business objects. For example, when you select the Customer complex type while running EMD, application-specific information will be added to the annotation of the CustomerWrapper and Customer business object definitions.

Custom business objects

If you use custom business objects, you must create predefined business objects using WebSphere Integration Developer business object wizard before running the enterprise service discovery wizard. The business object definitions created by the wizard are stored as xsd files on your local system. When the enterprise service discovery wizard creates business objects, it looks for the predefined business objects created in the business object wizard and populates them with data specific to the adapter project.

For more information on how to create predefined business objects, refer to the WebSphere Integration Developer documentation.

Enterprise service discovery

Use the enterprise service discovery wizard to generate business objects for an FTP server, capture the inbound and outbound connection properties, and generate the required artifacts (import, export, and wsdl files). The enterprise service discovery wizard provides a blueprint for business objects. In the enterprise service discovery wizard, you browse the metadata information of the FTP server, select the artifacts of interest, and generate service objects and descriptions that can be deployed. By selecting meta-object nodes from the metadata tree structure, you can generate business objects for the FTP server.

The metadata is transformed into service data objects consisting of business graphs and business objects. Through the enterprise service discovery wizard, you can perform the following actions:

- Generate business objects
- Set application-specific information on the business objects
- Set application-specific information on properties
- Provide service descriptions for inbound and outbound events
- Provide connection descriptions for inbound and outbound events

Globalization and bidirectional transformation

This 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 left-to-right (Hebrew or Arabic, for example) and right-to-left (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 Enterprise Service Bus use the Windows standard format, but an enterprise information system exchanging data with WebSphere Process Server or 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.

WebSphere Process Server 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.

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.

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
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.

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.

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.

Property type	Data transformations
Skip	Identifies content or metadata to exclude from transformation.
Special Format	Identifies certain text, such as file paths or URLs, that require 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 theWebSphere Process Server administrative console to configure these properties.
- **ManagedConnectionFactory (MCF) properties:** These properties are used at run time to create an outbound connection instance with an enterprise information system. Once the MCF 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 WebSphere Process Server administrative console.

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 (MCF) 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 5. Planning for adapter implementation

To implement the IBM WebSphere Adapter for FTP, you must plan for inbound and outbound processing and consider security and performance requirements.

Planning for inbound and outbound processing

For inbound and outbound communication, the adapter uses Commons Net API version 1.4.1 to access the remote file system. Commons Net API provides an interface that is used to parse the long list (ls -l) output for FTP servers, which deviates from the standard output. For more information about the Commons Net API version 1.4.1 libraries, see http://jakarta.apache.org/commons/net/.

Security

The adapter can use the SSL protocol to transfer data between the client application and the FTP server. For secure communication, a secure FTP server that supports the SSL protocol and contains a private key and certificate must be installed and configured.

Use the following information to implement secure FTP:

- The adapter uses a passive FTP mode of data transfer with a secure FTP server. If there is a firewall between the client and the server, the firewall settings might need to be configured to enable this mode.
- During SSL communication, the server sends its certificate to the client for verification. The client verifies the certificate to confirm that it is communicating with the intended server. To enable this verification process, the server's certificate must be present in the client's trust store. You can import the server's certificate into the client's trust store using the keytool utility. For example, enter the command `keytool -import -v -alias serverCert -file server.cert -keystore clientTrustStore`

  Where *server.cert* is the certificate of the server and *clientTrustStore* is the trust store of the client.
- Set the trust store by updating the JVM property. For example, enter the command *javax.net.ssl.trustStore=C:\MyKeyStore\clientTrustStore*

  Where *clientTrustStore* is the of the client specified in the previous step.
- Use FTPs as the protocol in the URL, to provide secure communication, for example, *ftps://host:port/ftpdir*.

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.

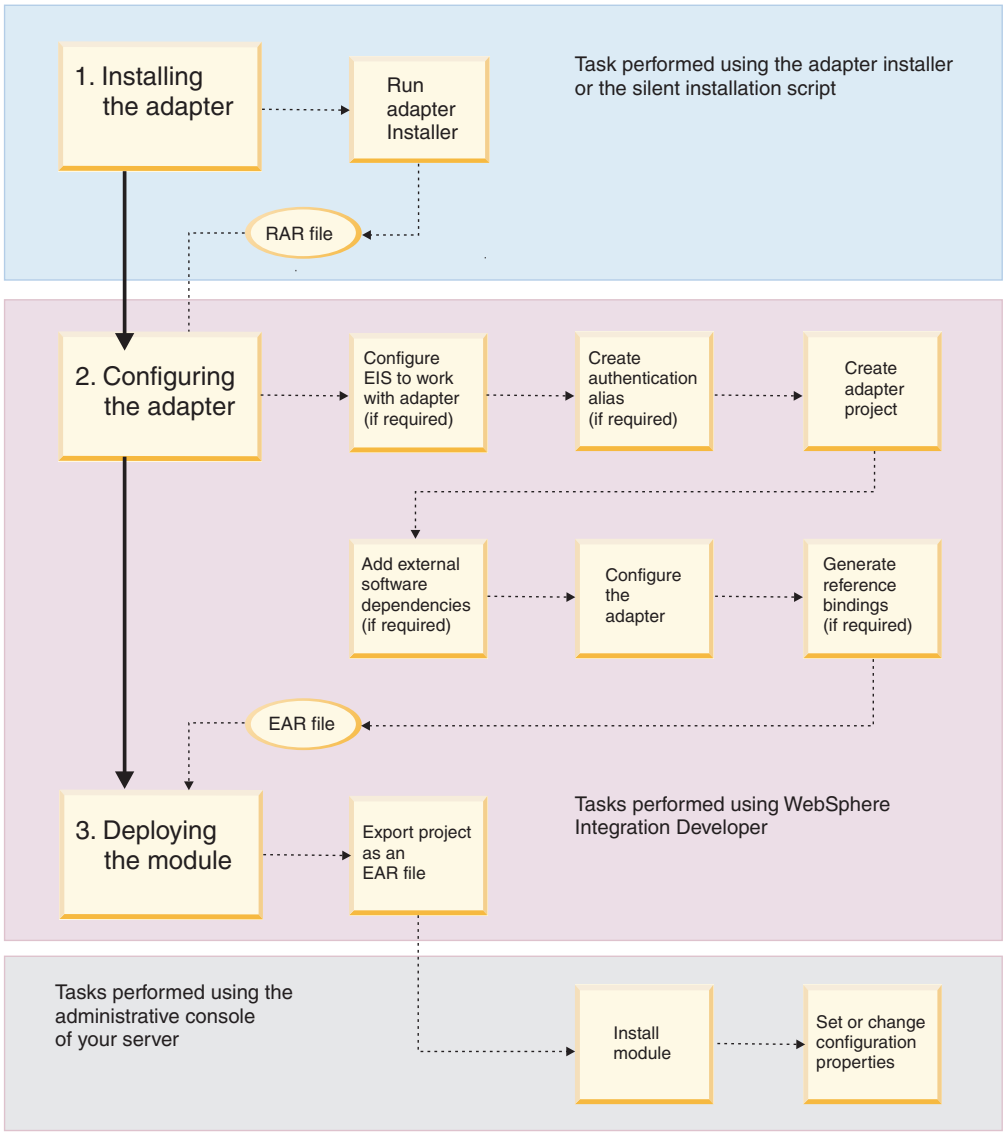


Figure 1. Roadmap for installing, configuring, and deploying the adapter

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 6. Installing the adapter

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

Installation prerequisites

Before installing Adapter for FTP, 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 FTP, Version 6.0.2.

Additional jar files

If you are using WebSphere Integration Developer version 6.0.1.1 or earlier, you must manually add three additional jar files to the classpath of the connector project. For more information on how to do this, see ″Adding jar files to WebSphere Integration Developer versions 6011 and earlier″ in the reference section.

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.

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.

   Note: Some WebSphere Adapters require you to perform additional steps specific to your adapter to complete the installation. The WebSphere Adapter for FTP does not have this requirement.
2. After performing the installation, 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\ftp\adapter\ftp\deploy\CWYFT_FTPFile.rar.

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 137.

Chapter 7. Configuring the adapter for deployment

To configure WebSphere Adapter for FTP 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 authentication alias

Create the authentication alias on the server by using the WebSphere Process Server administrative console. From the administrative console, configure the global security and set the password for the authentication alias, which is used to process both inbound and outbound requests.

About this task

To create an authentication alias, use the following procedure.

How to perform this task

1. On the WebSphere administrative console ″Welcome page,″ click **Security** → **Global security**.
2. Under the **Authentication** heading, click **JAAS Configuration** → **J2C Authentication data**.
3. Click **New**.
4. Type the required information in the **Alias**, **User ID**, **Password**, and **Description** fields.

   Note: The user ID and password that you type will be used to establish a connection to the enterprise information system for outbound processing.
5. Click **OK**, click **Save**, and then click **Save** again.

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.

Before you begin

You must have installed the Adapter for FTP and have created an authentication alias before proceeding.

About this task

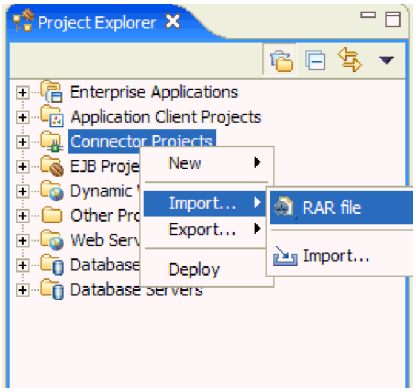
Create an adapter project (called a *connector project* in WebSphere Integration Developer) to contain the adapter (which you import from the adapter installation directory) as well as artifacts related to it. All projects are self-contained; they do not refer to objects outside of the project.

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 V6.0.2** → **WebSphere Integration Developer V6.0.2**.
   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**.

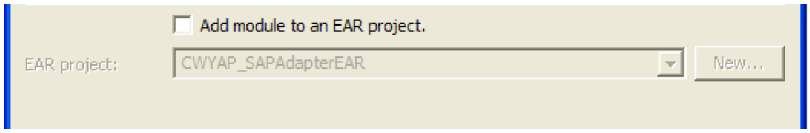
      If **J2EE** is not displayed in the Select Perspective window, select the **Show all** check box, click **J2EE**, and click **OK**.
   c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.
   d. Click **OK**.
3. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.



Import the RAR file navigation

4. From the Connector Import window, click **Browse** and navigate to the directory in which Adapter for FTP Software was installed.
5. Click **CWYFT_FTPFile.rar**.

   The connector project has the same name as the RAR file.
6. **Optional:** In the **Connector project** field, either type another name for the project or accept the default value.
7. **Optional:** In the **Target server** field, either 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.



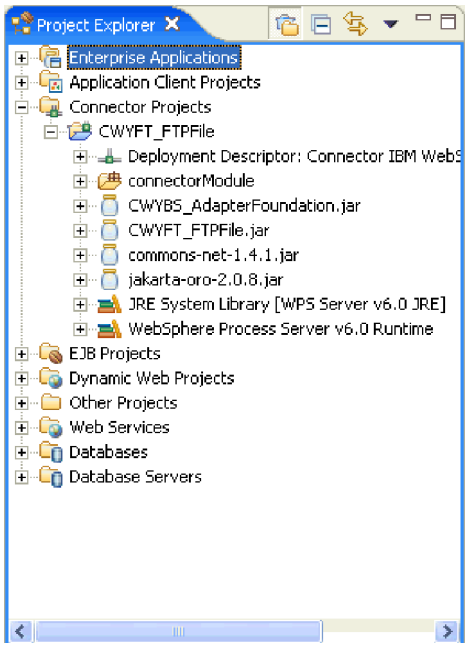
Add module to an EAR project check box

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 CWYFT_FTPFile, expand **CWYFT_FTPFile**.



CWYFT_FTPFile project window

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 commonj.connector.runtime.RecordHolderDataBinding interface. This interface is located in the commonj.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.

Configuring the adapter for outbound processing

To configure WebSphere Adapter for FTP 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.

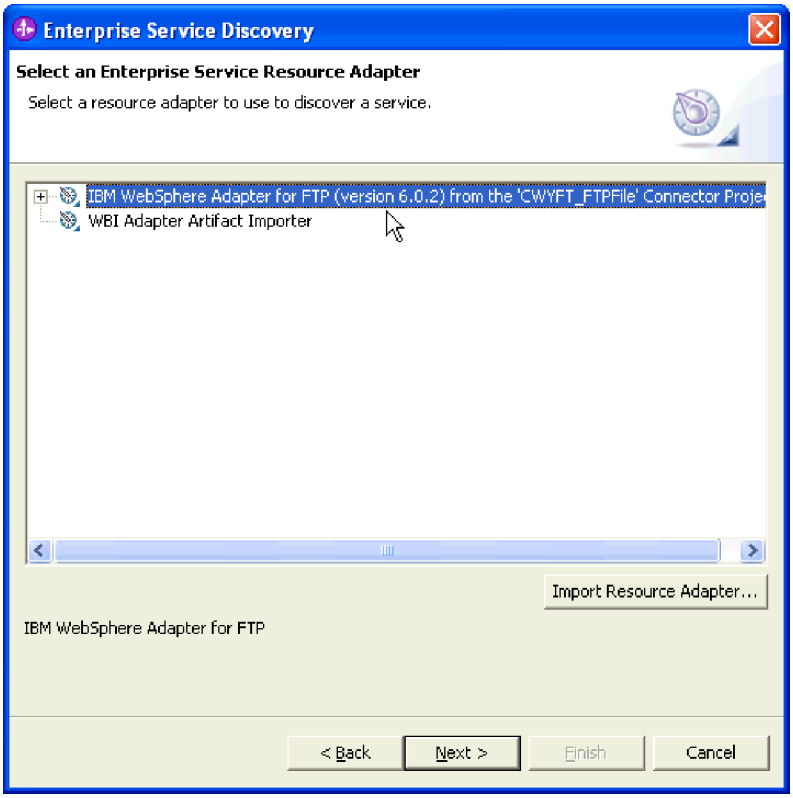
Generating business objects using enterprise service discovery

Use the enterprise service discovery wizard to identify business objects and services that can be used for outbound communication with the FTP server.

Setting connection properties for enterprise service discovery

Use enterprise service discovery (ESD) to view all services available to the adapter and to configure your FTP server connection settings.

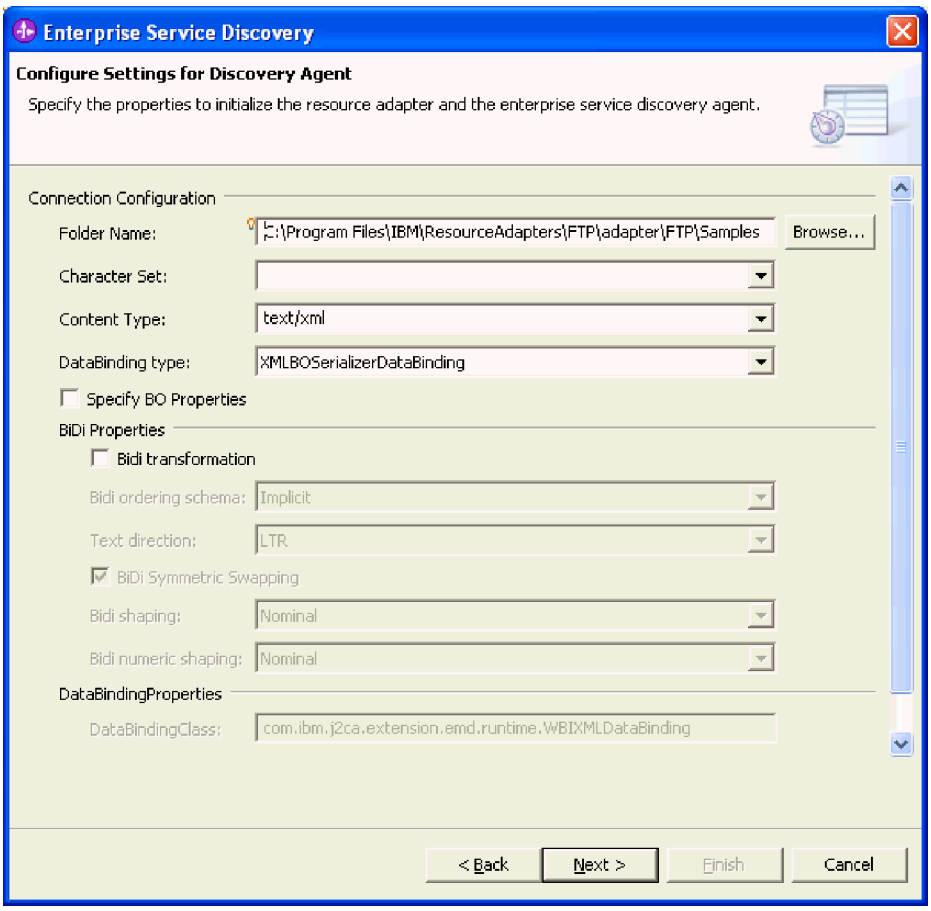
1. From the WebSphere Integration Developer window, switch to the Business Integration perspective.
   a. Select **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed.
   b. Select the **Business Integration** perspective.
2. Right-click the frame of the Business 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. Select **IBM WebSphere Adapter for FTP** from the **Import Configurations** menu and then click **Next**.



Enterprise Service Resource Adapter window

4. In the Configure settings for Discovery Agent window, specify the properties that are used to discover the business data as well as for selecting the data binding that is used at runtime.
   a. Enter the **Folder name** for where your xsd schemas for the business objects have been stored. For example: `C:\Program Files\IBM\ResourceAdapters\FTP\adapter\FTP\Samples`. The business objects to be used in the integration scenarios are selected from schema definitions present in the folder.
   b. **Optional:** Select a **Character Set**. Select the Character Set if the business data is of a different encoding. The business data corresponds to the data present in the files on which the operations are performed.

      Note: If using the samples scenarios generated by WebSphere Integration Developer, you do not need to select a Character Set.
   c. Enter **Content type**. This one-time setting is used to bundle a content-type with a corresponding data binding. This displays all the content types that the adapter supports.
   d. **Optional:** If you want to configure individual business object properties, select the **Specify BO Properties** check box. If checked, a series of windows will appear where individual business object properties can be defined.
5. When all properties are defined, click **Next**.



Configure Settings for Discovery Agent window

Result

Using these properties, the discovery service prepares a metadata tree that you use to select objects and navigate subsequent tasks.

Selecting business objects and services

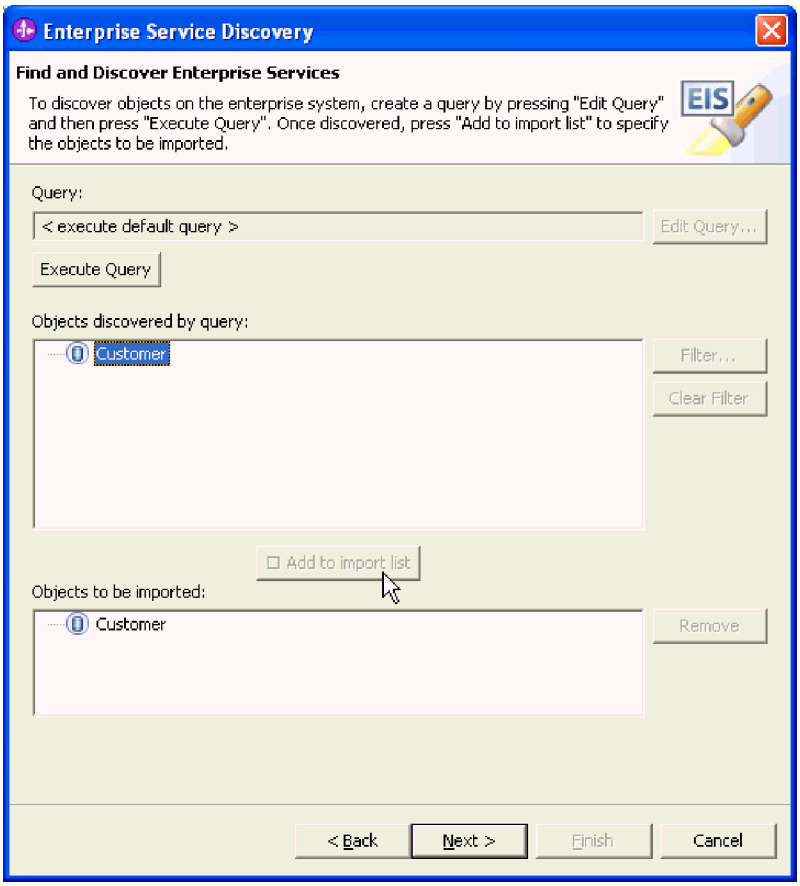
To configure how the adapter will function after it is deployed to the server, use the enterprise service discovery wizard to select business objects and services for use with your adapter.

Before you begin

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

How to perform this task

1. In the Find and Discover Enterprise Services window, select **Execute Query** to browse for business objects.
2. Select business objects from the metadata objects tree.
   a. Select the business objects you want to add to your connector project.
   b. Click **Add to import list** to add your business objects to the project This button is enabled only for objects that can be imported. For all other objects, the button is disabled.



Find and Discover Enterprise Services window

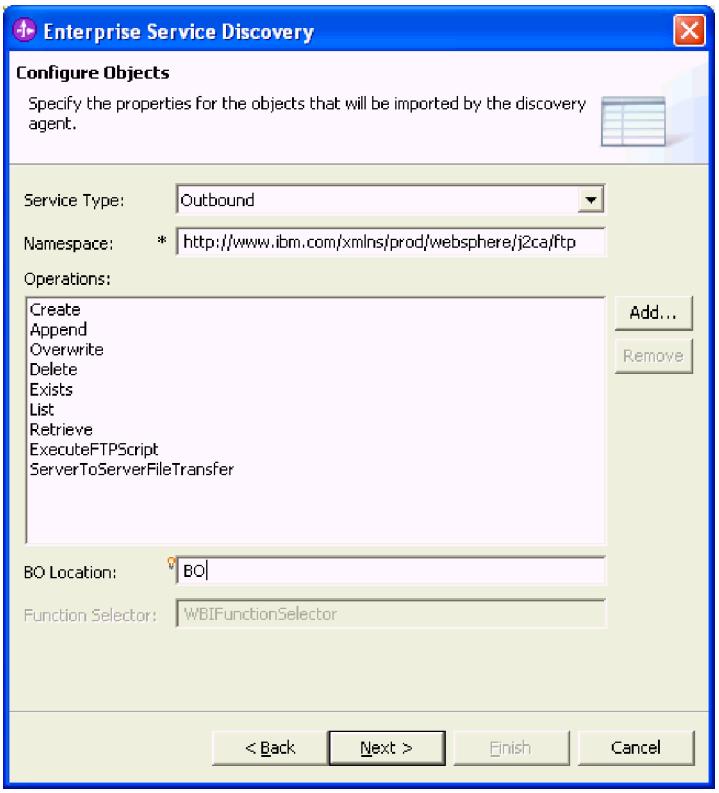
Objects are shown in the Objects to be imported window. If you want to remove them, select them and click the **Remove selected** button.

3. **Optional:** If you selected to configure additional business object properties on the Configure Settings for Discovery Agent window, the Configuration Parameters for Customer window will open for you to specify your individual business object properties for each business object when you click the **Add to import list** button.
4. When all of the desired business objects are added to the project, 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.



Configure objects window

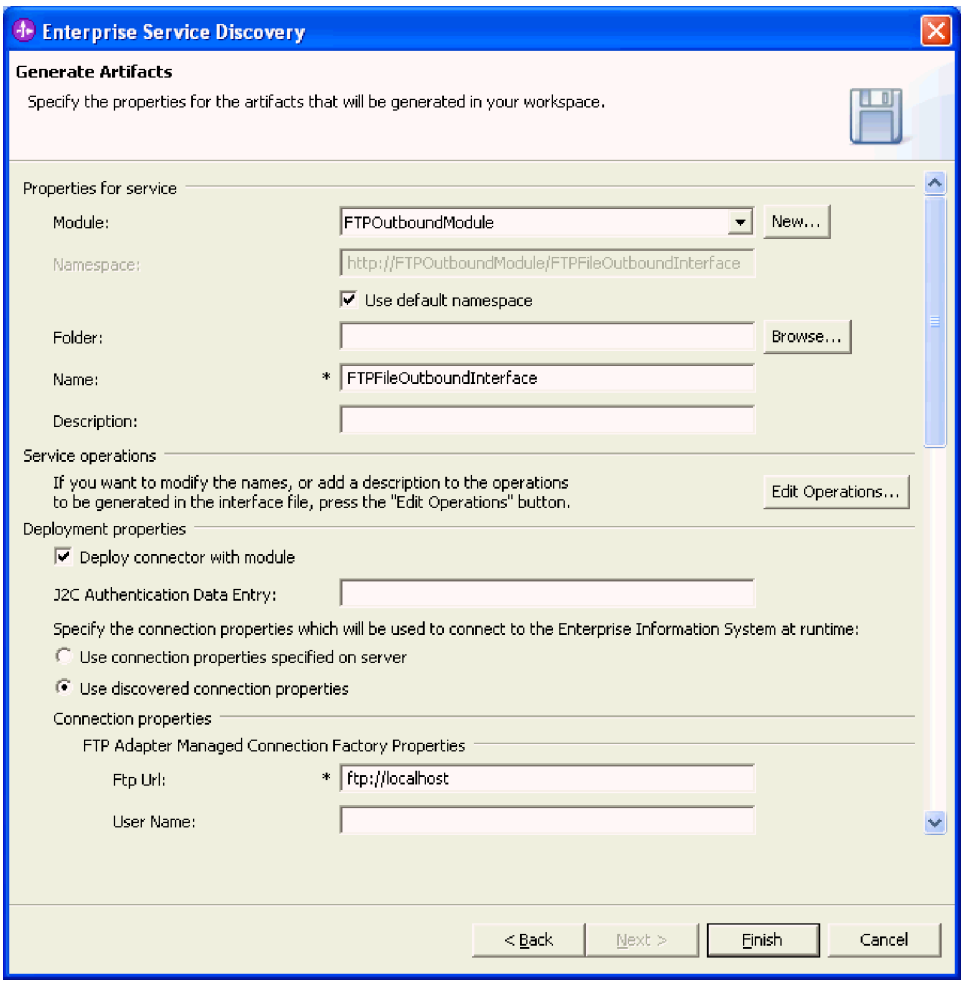
Result

The objects are now configured for outbound communication.

Generating artifacts

Generate business object definitions and related artifacts by using the enterprise service discovery wizard to first add a container business object to the business function and then create a new assembled adapter application, also referred to as an SCA module. After the business object definitions and related artifacts are generated, they are contained within the newly assembled adapter application (the SCA module).

1. In the Generate Artifacts window, click **New** next to the **Module** field to create a new module.
2. Select **Create a module project** and click **Next**.
3. In the New Module window, type a meaningful name in the **Module Name** field, then click **Finish**. The name you type is the name that will be given to the assembled adapter application (the SCA module) after the business object is generated.
4. In the Generate Artifacts window, select the **Use discovered connection properties** option.



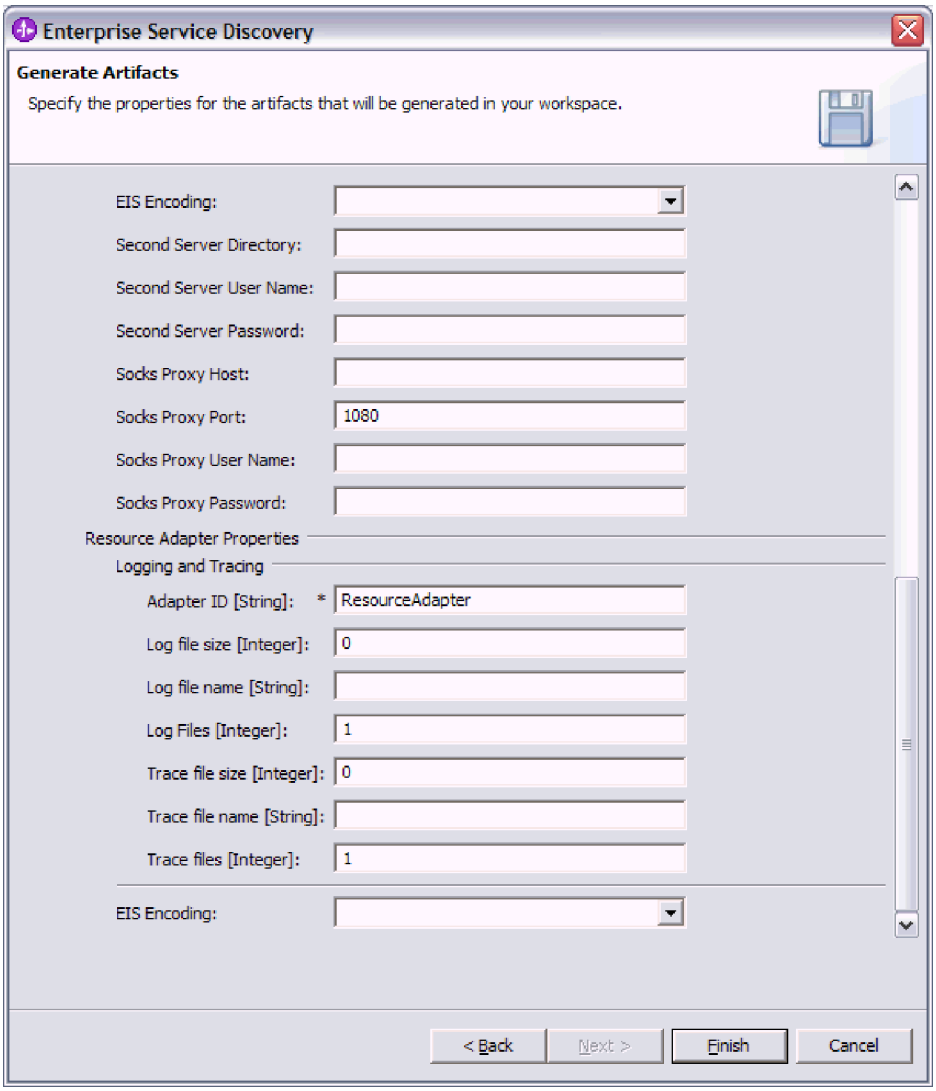
Generate Artifacts window

5. Enter the FTP Adapter Managed Connection Factory Properties. See "Managed (J2C) connection factory properties" on page 120 for more information on the properties. You must populate the **Ftp Url**, **User Name** and **Password** fields. The **Ftp Url** can be provided in one of the following formats:
   - `ftp://hostname[:portNumber]`
   - `ftp://[username:password@]hostname[:portNumber]`

     If using this format, you do not need to enter **User Name** and **Password** fields again.

     Note: The `portNumber` is optional in both of the above options.
   - `ftp://hostName`

     If using this format, provide the authentication alias by entering the value of the alias name in the **J2C Authentication Data Entry** property.
6. **Optional:** Scroll down to enter the logging and tracing properties.



Lower half of the Generate Artifacts window

7. Click **Finish**.

Result

The WSDL, import, FTPFileBG, FTPFile, 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

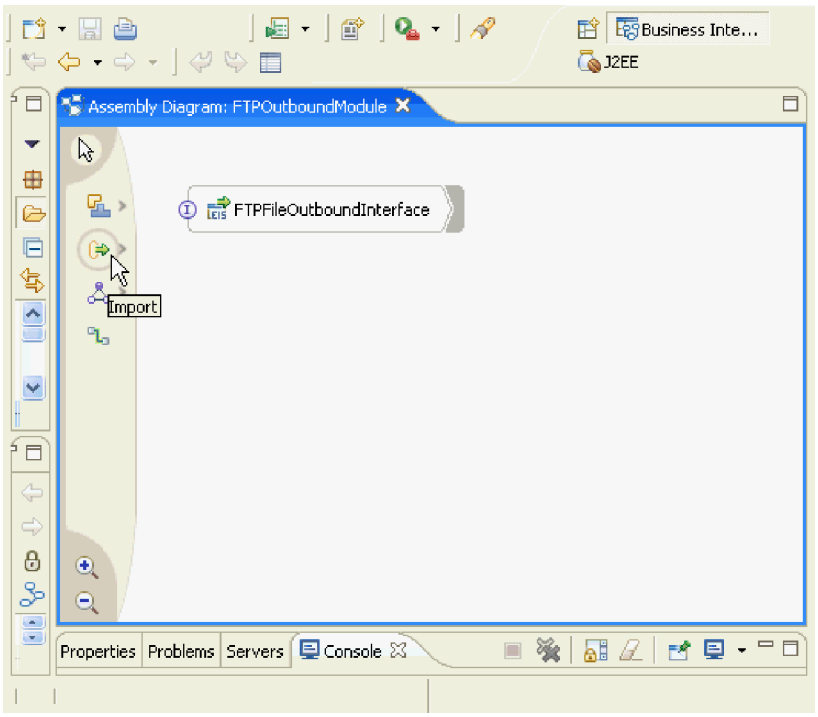
Create a reference in the assembly editor from the adapter project to a stand-alone reference. The stand-alone reference represents a generic J2EE componentr. By wiring the adapter project to the stand-alone reference, you link the adapter to other server processes.

Before you begin

You must have already saved and configured your adapter project before creating a reference binding.

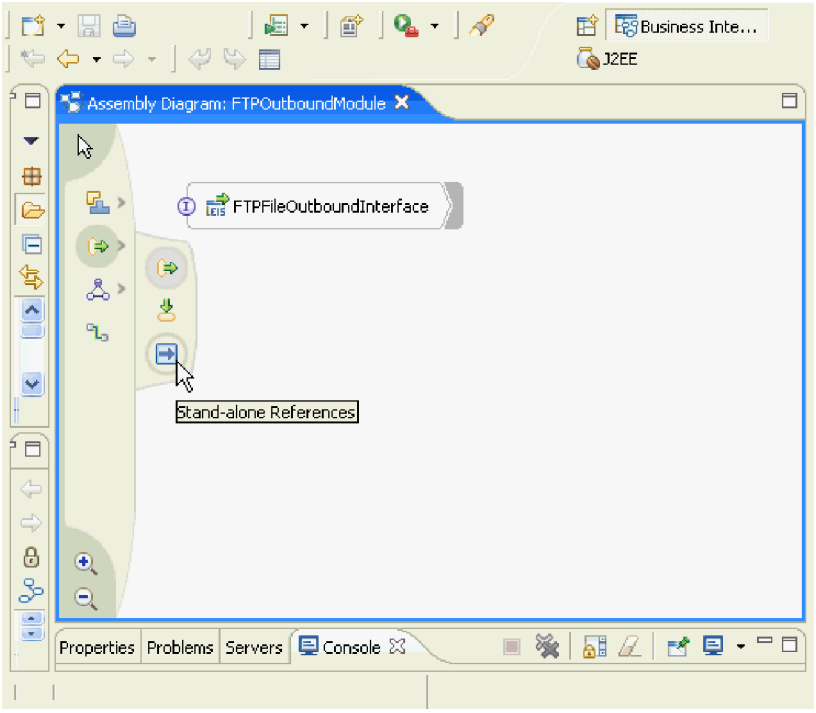
How to perform this task

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 module and click **Open With** → **Assembly Editor**. The Assembly Diagram window is displayed with the modules Import component in view.
3. Create a Stand-alone reference:
   a. Click the **Import** icon in the left-side (vertical) frame of the Assembly Diagram window to display the icons.



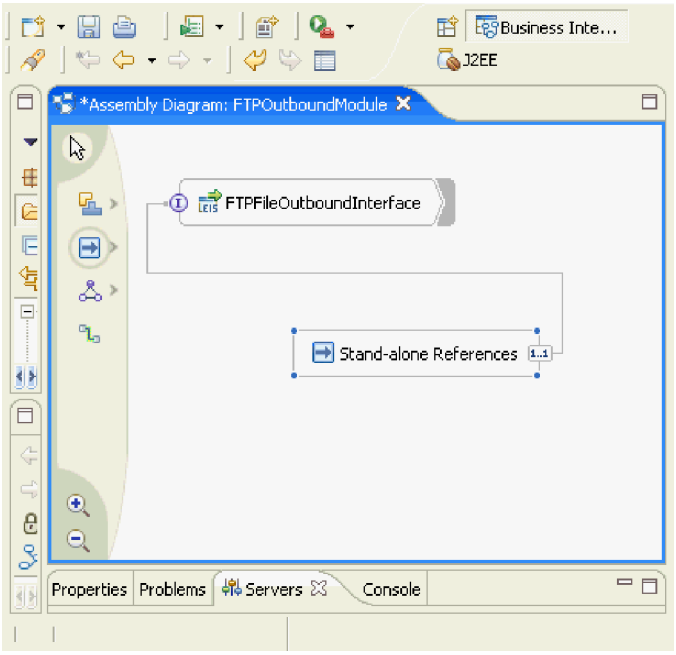
The Import icon in the Assembly Diagram window

   b. Drag the **Stand-alone Reference** icon into the Assembly Diagram window.



Stand-alone Reference icon

c. Create a wire from the Stand-alone Reference to the adapter project by clicking the side of the Stand-alone references representation and dragging the wire to the adapter project representation.



Component wiring window

d. Click **File** → **Save**.

4. Click **File** → **Save**.

Result

The reference bindings are generated.

Configuring the adapter for inbound processing

To configure WebSphere Adapter for FTP 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.

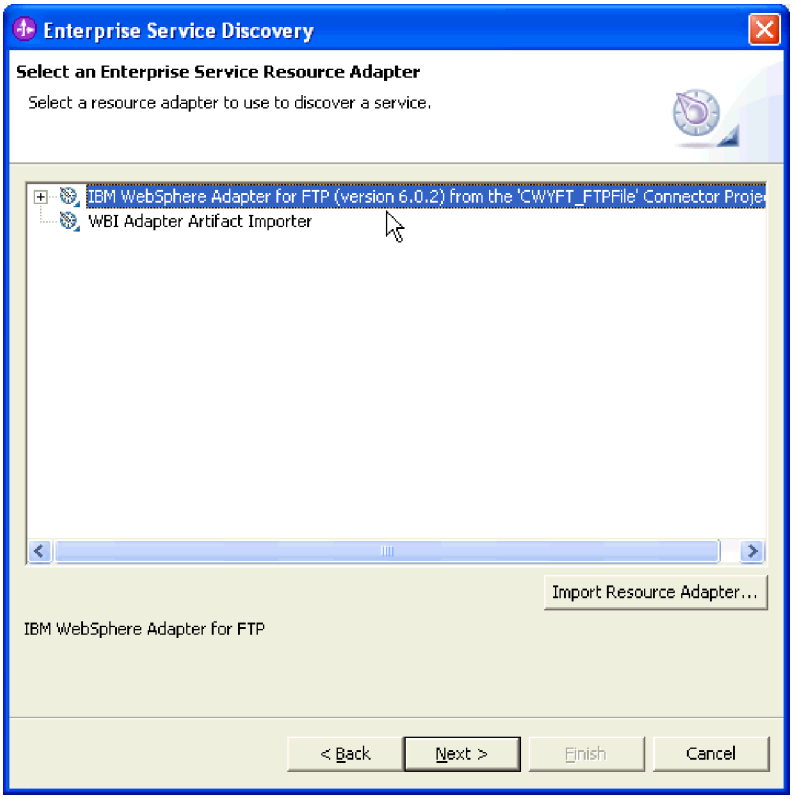
Generating business objects using enterprise service discovery

Use enterprise service discovery to identify business objects and services that can be used for inbound communication with the FTP server.

Setting connection properties for enterprise service discovery

Use enterprise service discovery (ESD) to view all services available to the adapter and configure your FTP server connection settings.

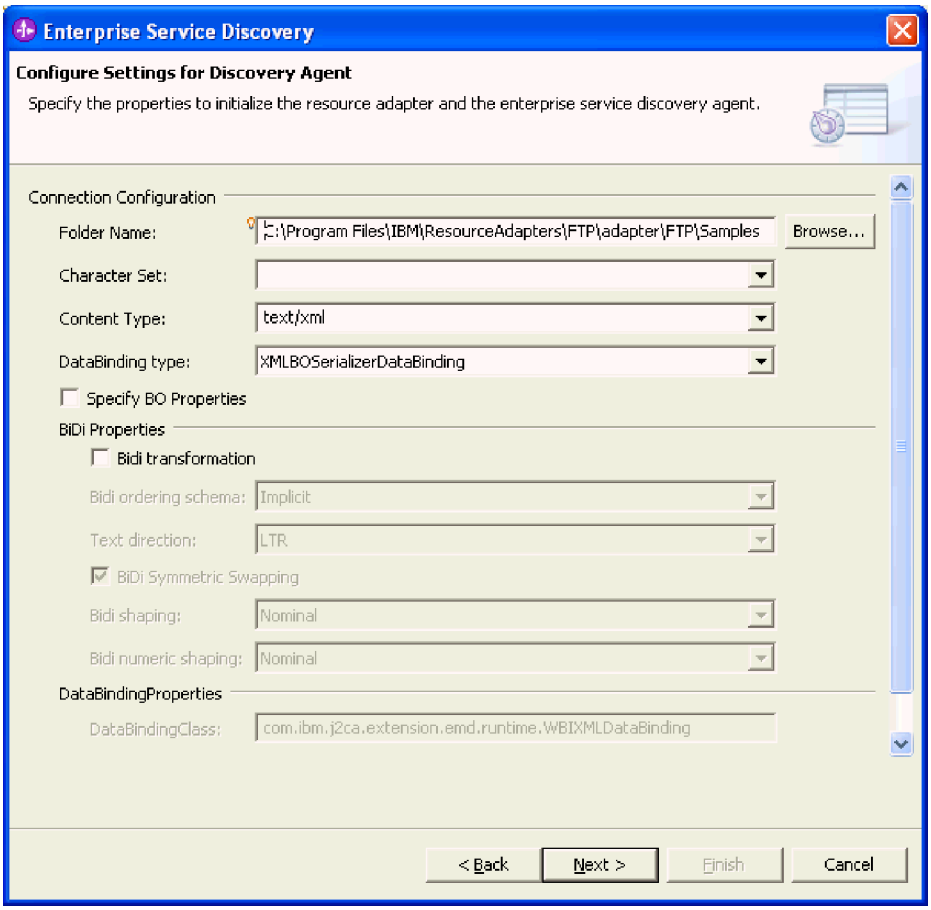
1. From the WebSphere Integration Developer window, switch to the Business Integration perspective.
   a. Select **Window** → **Open Perspective** → **Other** from the menu bar. All perspectives are displayed.
   b. Select the **Business Integration** perspective.
2. Right-click the frame of the Business 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. Select **IBM WebSphere Adapter for FTP** from the **Import Configurations** menu and then click **Next**.



Enterprise Service Resource Adapter window

4. In the Configure settings for Discovery Agent window, specify the properties that are used to discover the business data as well as for selecting the data binding that is used at runtime.
   a. Enter the **Folder name** for where your xsd schemas for the business objects have been stored. For example: `C:\Program Files\IBM\ResourceAdapters\FTP\adapter\FTP\Samples`. The business objects to be used in the integration scenarios are selected from schema definitions present in the folder.
   b. **Optional:** Select a **Character Set**. Select the Character Set if the business data is of a different encoding. The business data corresponds to the data present in the files on which the operations are performed.

      Note: If using the samples scenarios generated by WebSphere Integration Developer, you do not need to select a Character Set.
   c. Enter **Content type**. This one-time setting is used to bundle a content-type with a corresponding data binding. This displays all the content types that the adapter supports.
   d. **Optional:** If you want to configure individual business object properties, select the **Specify BO Properties** check box. If checked, a series of windows will appear where individual business object properties can be defined.
5. When all properties are defined, click **Next**.



Configure Settings for Discovery Agent window

Result

Using these properties, the discovery service prepares a metadata tree that you use to select objects and navigate subsequent tasks.

Selecting business objects and services

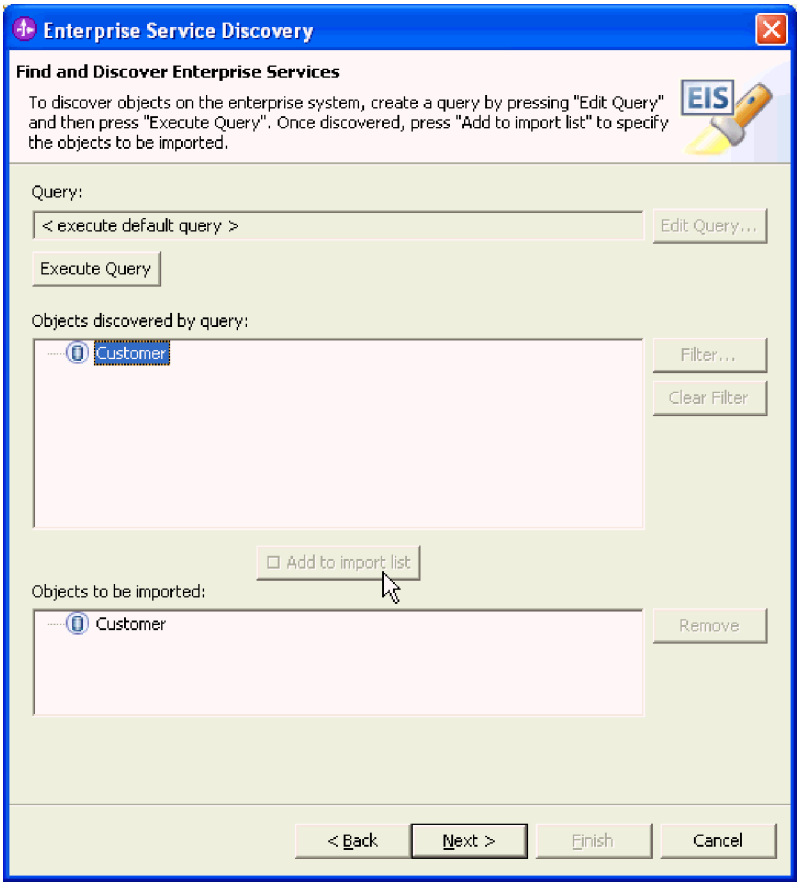
Use the enterprise service discovery wizard to select business objects and services for use with your adapter.

Before you begin

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

How to perform this task

1. In the Find and Discover Enterprise Services window, select **Execute Query** to browse for business objects.
2. Select business objects from the metadata objects tree.
   a. Select the business objects you want to add to your connector project.
   b. Click **Add to import list** to add your business objects to the project. This button is enabled only for objects that can be imported. For all other objects, the button is disabled.



Find and Discover Enterprise Services window

Objects are shown in the Objects to be imported window. If you want to remove them, select them and click **Remove selected**.

3. **Optional:** If you selected to configure additional business object properties on the Configure Settings for Discovery Agent window, the Configuration

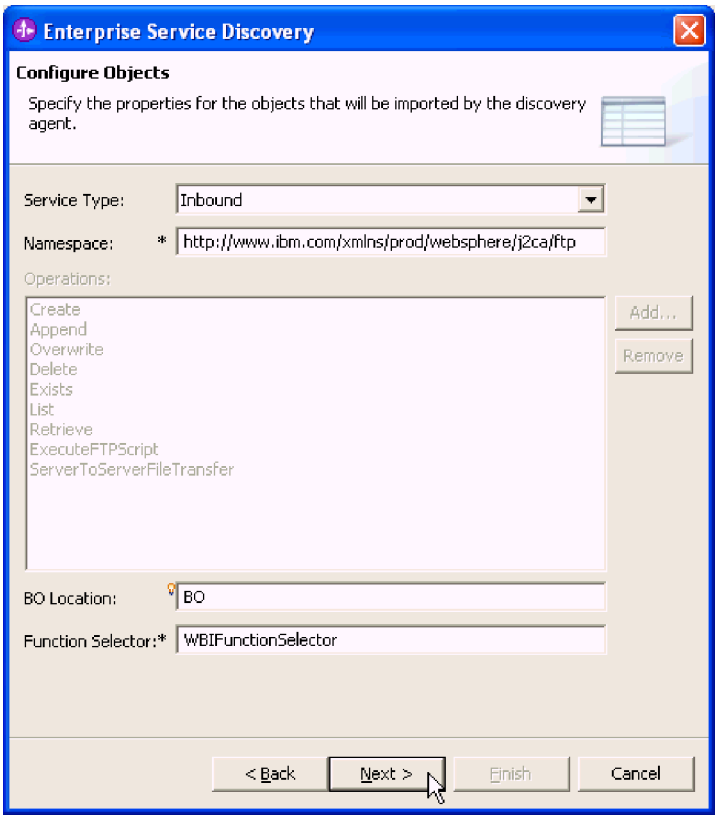
Parameters for Customer window will open for you to specify your individual business object properties for each business object when you click the **Add to import list** button.

4. When all of the desired business objects are added to the project, 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 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.



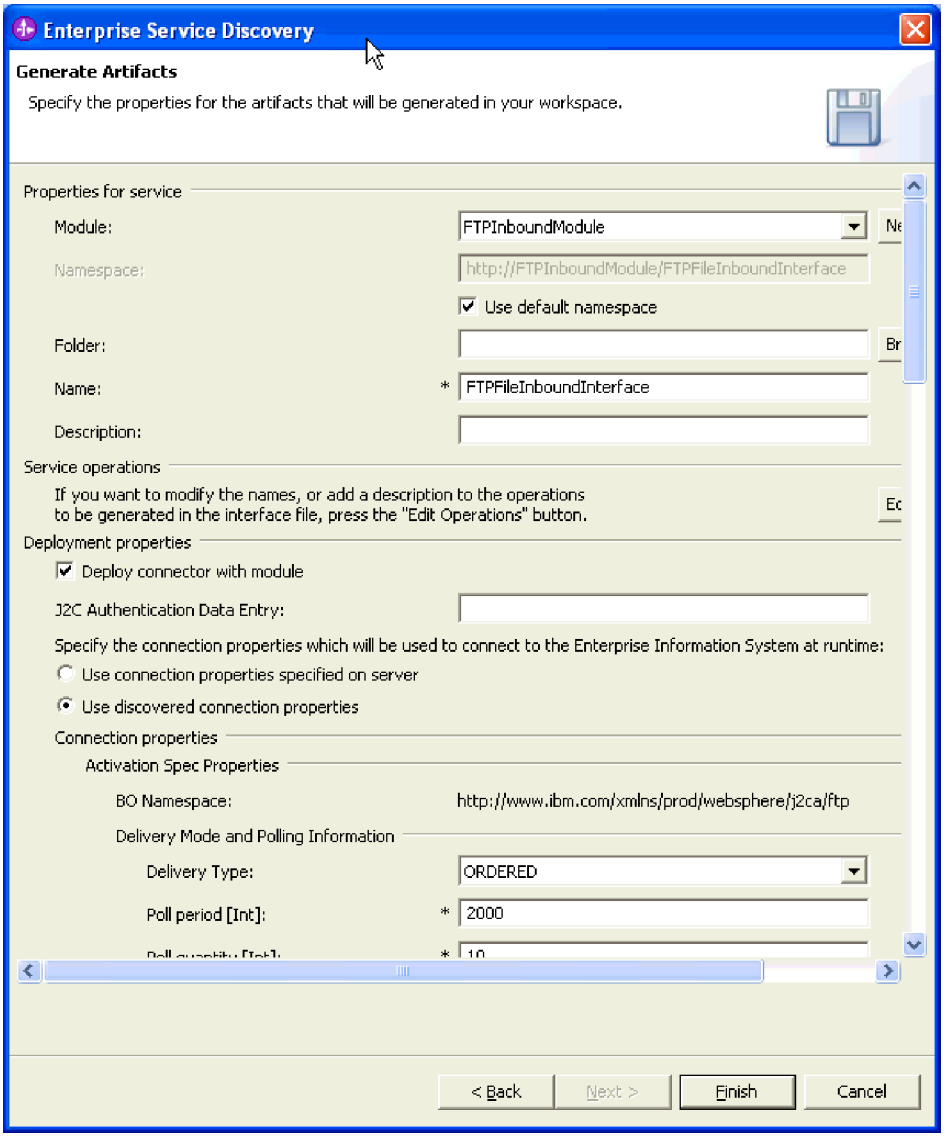
Configure objects window

Generating artifacts

Generate business object definitions and related artifacts by using the enterprise service discovery wizard to first add a container business object to the business function and then create a new assembled adapter application, also referred to as an SCA module. After the business object definitions and related artifacts are generated, they are contained within the newly assembled adapter application (the SCA module).

1. In the Generate Artifacts window, click **New** next to the **Module** field to create a new module.

2. In the New Module window, type a meaningful name in the **Module Name** field, then click **Finish**. The name you type is the name that will be given to the assembled adapter application (the SCA module) after the business object is generated.
3. In the Generate Artifacts window, select the **Use discovered connection properties** option.



Generate Artifacts window

4. Scroll down to enter the required Activation Spec Properties, as indicated by the asterisk (*). See "Activation specification properties" on page 124 for more information on the properties. These required properties are:
   - **DataSource JNDI Name**

     The JNDI name used to create a JDBC connection from a data source that will be used by the adapter to store the events in the event persistence database table. The data source must already be configured in WebSphere Process Server's administrative console.
   - **Event Table Name**

     Table name that is created to store the events.
   - **Event Directory**

`ftp://[username:password@]hostName[:portNumber]/eventDirectory`

If username and password are not specified here, they must be specified in the **User Name** and **Password** fields.

- **Event File Mask**

  Polls only the files that match the file mask, which by default is `*.*`.
- **FTP Get Quantity**

  Number of files retrieved from the remote FTP URL with each remote poll.
- **FTP Poll Frequency**

  Determines the frequency that the adapter polls the FTP server. For example, if set to `6`, the adapter will poll the event directory once every 6 standard polls.
- **Local Event Directory**

  Specifies the local system directory where the adapter will download the event files from the FTP server.

5. **Optional:** Scroll down to enter the logging and tracing properties.
6. Click **Finish**.

Result

The WSDL, import, FTPFileBG, FTPFile, 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

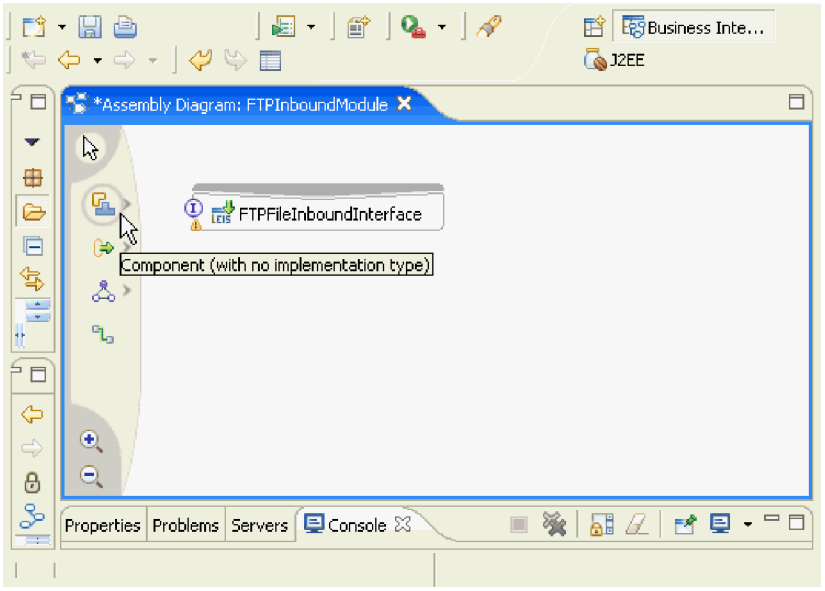
Create a reference binding to an adapter from the project module to link the adapter to other server processes.

Before you begin

You must have already saved and configured your adapter project before creating a reference binding.

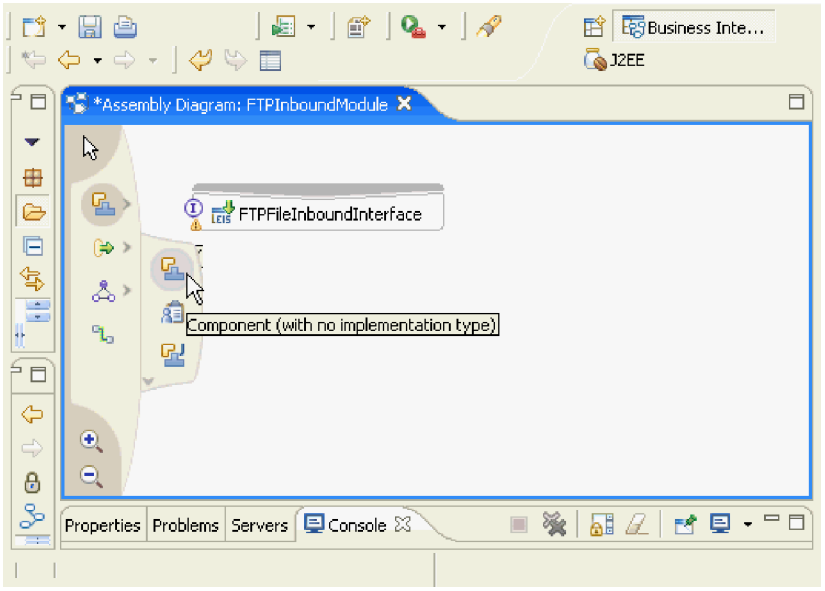
How to perform this task

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 module and click **Open With** → **Assembly Editor**.
   a. Click the **Component (with no implementation type)** icon.



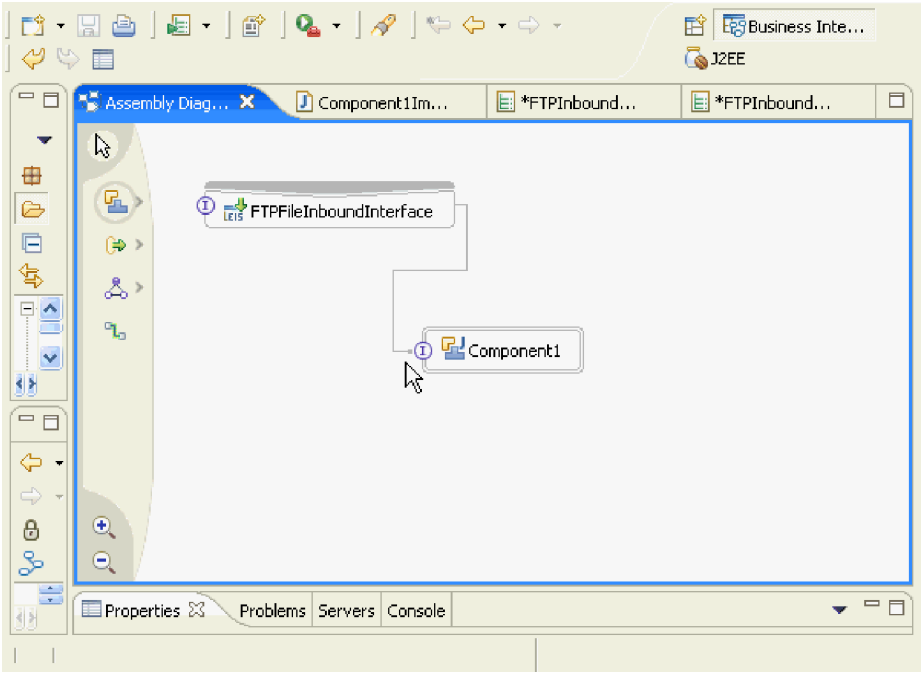
The Component icon in the Assembly Diagram window

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



Component icon

c. Click the **Wire** icon and drag the wire from **FTPInboundInterface** to **Component1** to create the wiring.



Component wiring window

d. Right-click **Component1** and select **Generate Implementation** → **Java**.
e. Select **(default package)** and click **OK**.

3. Click **File** → **Save**.

Result

The reference bindings are generated.

Chapter 8. Deploying the module

To deploy the adapter project to the application server, export the project as an enterprise archive file, install the adapter project, add configuration properties that were not set in the enterprise service discovery wizard, and then, if necessary, cluster your adapter project applications.

Exporting the project as an 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 the WebSphere Integration Developer window, right-click the module and select **Export** from the pop-up menu. The Export window is displayed.
2. Select **EAR file** from the Export - Select window. The Export - EAR Export window is displayed.
3. In the EAR Export window, select the EAR project and the destination directory. The destination directory is the directory, including the EAR file name, to which the project must be exported.
4. Click **Finish**.

Result

The adapter project is exported to an EAR file.

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

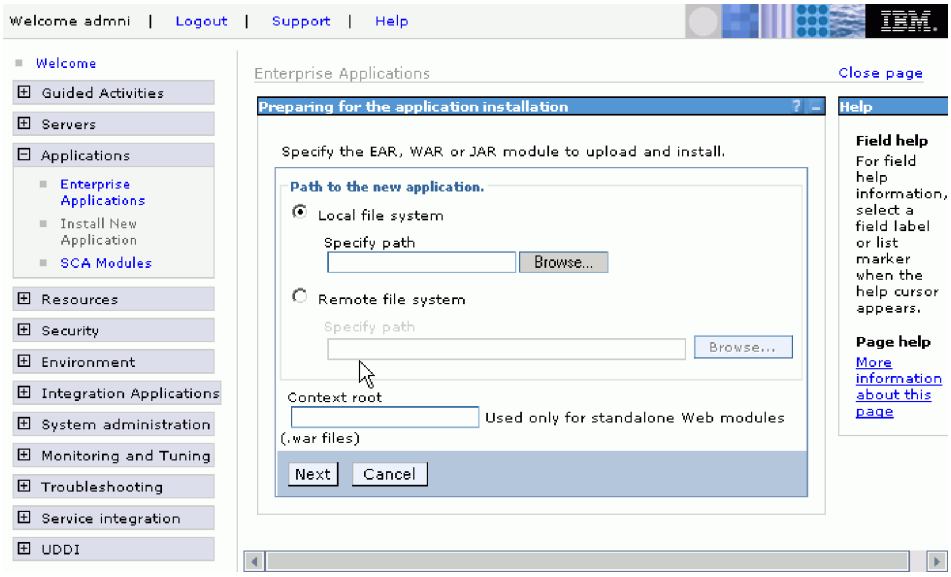
You must have exported your project module as an EAR file before installing the adapter project.

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.

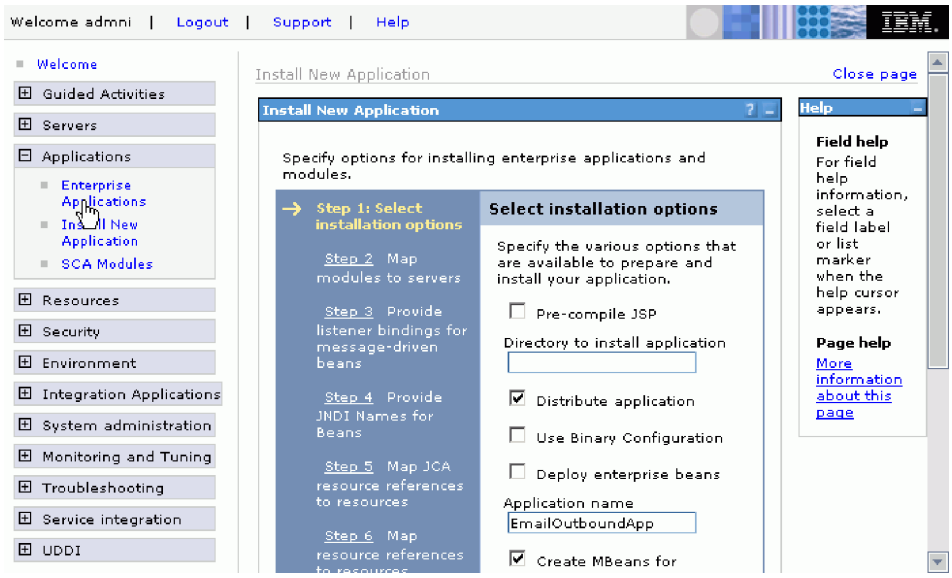
How to perform this task

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 configuration properties from the administrative console

Configuration properties can be set using the enterprise service discovery wizard or the administrative console of your server. These topics describe how to change the configuration properties using the administrative console.

Setting resource adapter properties

You can set the resource adapter properties using the administrative console.

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**

3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Click **CWYFT_FTPFile.rar**.
6. Click **Resource Adapter**.
7. Under **Additional Properties**, select **Custom properties**.
8. 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.

   See "Resource adapter properties" on page 119 for more information on the properties.
9. Click **Save** in the **Messages** box at the top of the window.

Setting managed (J2C) connection factory properties

To reset a managed connection factory property or to add a new value, use the administrative console. Managed connection factory configuration properties are used at run time to create an outbound connection instance with an enterprise information system.

How to perform this task

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**
3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Click **CWYFT_FTPFile.rar**.
6. Click **Resource Adapter**.
7. Under **Additional Properties**, select **J2C connection factories**.
8. Click the name of the J2C connection factory you want to configure.
9. Click **Custom properties**. Custom properties are those J2C connection factory properties that are unique to Adapter for FTP. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
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.

    See "Managed (J2C) connection factory properties" on page 120 for more information on the properties.
11. Click **Save** in the **Messages** box at the top of the window.

Setting activation specification properties

To reset an activation specification property or to add a new property value, use the administrative console. Activation specification properties hold the inbound event processing configuration information for a file endpoint.

How to perform this task

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**
3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Click **CWYFT_FTPFile.rar**.
6. Click **Resource Adapter**.
7. Under **Additional Properties**, select **J2C Activation specifications**.
8. Click the name of the adapter application you want to configure.
9. Click **J2C activation specification custom properties**.
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.

    See "Activation specification properties" on page 124 for more information on the properties.
11. Click **Save** in the **Messages** box at the top of the window.

Chapter 9. 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 10. 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.

Troubleshooting event processing

If the database server goes down during an inbound operation, the adapter sends an exception message. The database server and the adapter must be restarted to process the events.

The adapter will automatically start processing the events when the database and itself are restarted.

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 FTP. 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/supp 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 1. 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 11. Quick start tutorials

The tutorials in this section demonstrate how WebSphere Adapter for FTP 6.0.2 performs inbound and outbound operations.

Introduction

Each tutorial provides a complete set of instructions for configuring the adapter so that it can be used by WebSphere Process Server or WebSphere Enterprise Service Bus to send requests to the FTP server or by the FTP server to send requests to a WebSphere Process Server or WebSphere Enterprise Service Bus.

Learning objectives

The learning objectives for this tutorial are to configure, deploy, and test the adapter for the following scenarios:

1. Outbound processing with data transformation for the create operation
2. Outbound processing with data transformation for the append operation
3. Outbound for the delete operation
4. Outbound for the retrieve operation
5. Outbound for the list operation
6. Inbound processing with data transformation
7. Inbound processing with simple pass through

Time required

Each tutorial should take approximately twenty minutes to complete.

Audience

This tutorial is 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:

- WebSphere Integration Developer, version 6.0.2
- IBM WebSphere Adapter for FTP version 6.0.2

Accessing the tutorial files

The tutorial files are replicas of the artifacts that you create with the enterprise service discovery wizard in these scenarios, and are available for reference so that you can verify that the files that you create with the enterprise service discovery are correct.

1. Run the installer to install the RAR file (which contains the adapter) and the samples folder.
2. Extract the contents of the emdsample.zip file located in the samples\referencefiles folder.

Result

A set of files is extracted into the *emdsample* subfolder, which is a user-defined name for the directory where you will save the extracted files.

Scenario or usage example.

Business objects and artifacts used in the tutorials

Business objects and artifacts that are specific to either outbound or inbound processing are described below.

Business objects and artifacts used in the outbound scenario

File name	Description
FTPOutboundModule/BO/Customer.xsd	Definition for the Customer business object
FTPOutboundModule/BO/CustomerWrapper.xsd	Definition for the business object container
FTPOutboundModule/BO/CustomerWrapperBG.xsd	Definition for the business object graph
FTPOutboundModule/BO/FTPFile.xsd	Definition for the general business object
FTPOutboundModule/BO/FTPFileBG.xsd	Definition for the general business object graph
FTPOutboundModule/BO/UnstructuredContent.xsd	Definition for the general business object
FTPOutboundModule/BO/FileContent.xsd	Definition for the general business object
FTPOutboundModule/BO/ListResponse.xsd	Definition for the list business object
FTPOutboundModule/BO/ListResponseBG.xsd	Definition for the list business object graph
FTPOutboundModule /BO/ RetrieveResponseWrapper.xsd	Definition for the retrieve business object
FTPOutboundModule/BO/ RetrieveResponseWrapperBG.xsd	Definition for the retrieve business object graph
FTPOutboundModule /BO/ExistsResponse.xsd	Definition for the exists business object
FTPOutboundModule /BO/ExistsResponseBG.xsd	Definition for the exists business object graph
FTPOutboundModule/FTPFileOutboundInterface.import	Contains the SCA import for the adapter
FTPOutboundModule /FTPFileOutboundInterface.wsdl	Service interface to invoke the adapter

Business objects and artifacts used in the inbound scenario

File name	Description
FTPInboundModule/BO/Customer.xsd	Definition for the Customer business object
FTPInboundModule/BO/CustomerWrapper.xsd	Definition for the business object container
FTPInboundModule/BO/CustomerWrapperBG.xsd	Definition for the business object graph
FTPInboundModule/BO/FTPFile.xsd	Definition for the general business object
FTPInboundModule/BO/FTPFileBG.xsd	Definition for the general business object graph
FTPInboundModule/BO/UnstructuredContent.xsd	Definition for the general business object
FTPInboundModule/FTPFileInboundInterface.export	Contains the SCA import for the adapter
FTPInboundModule/FTPFileInboundInterface.wsdl	Service interface to invoke the adapter

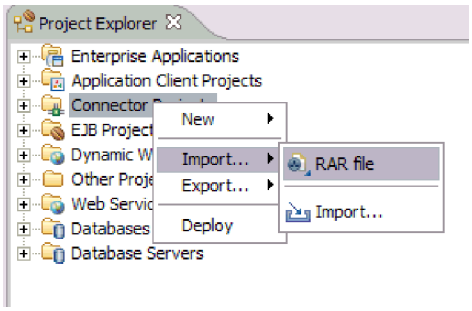
Tutorial: Sending outbound data to the FTP server

The outbound tutorials demonstrate data transformation for both the create and append operations, and basic delete, retrieve, and list operations for data being sent to the FTP server.

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. Click **Start** → **Programs** → **IBM WebSphere** → **Integration Developer 6.0** to Launch WebSphere Integration Developer.
2. Click **Window** → **Open perspective** → **Other** to switch to the J2EE perspective.
3. Right-click the Connector Projects folder, then select **Import** → **RAR file** .

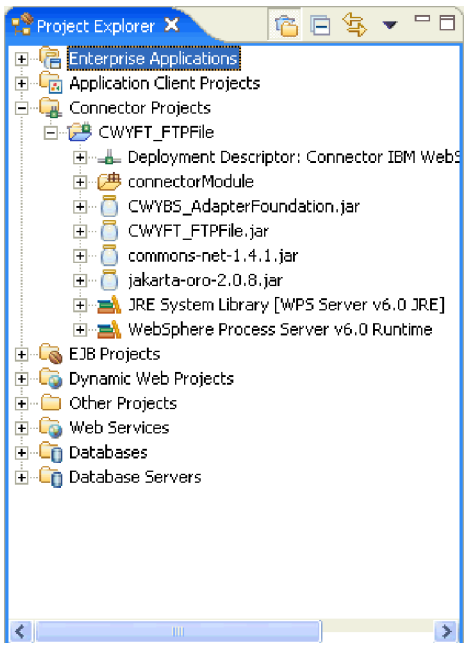


Import the RAR file navigation

4. In the Connector import window, click **Browse**, and select the RAR file location.
5. In the **Connector project** field, specify a project name.
6. Clear the **Add module to an EAR project** check box.
7. Accept all other defaults.
8. Click **Finish**.

Result

The RAR file is imported and a connector project is created in the workspace.



CWYFT_FTPFile project window

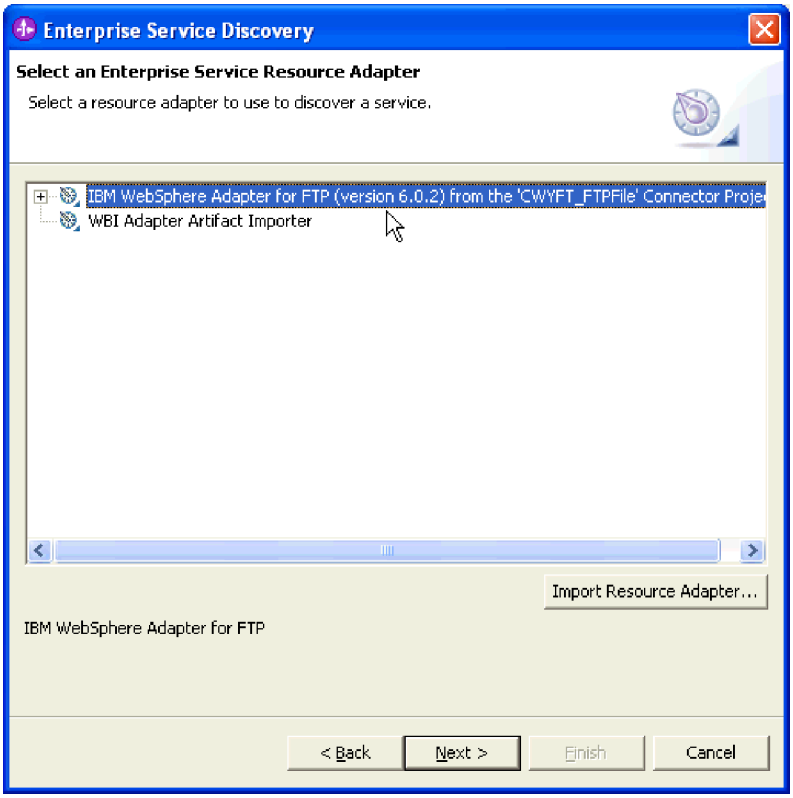
Configuring the adapter for outbound processing

Use the enterprise service discovery wizard to set the adapter configuration properties and to generate the necessary business objects to use for outbound processing.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to view all services available to the adapter and configure your FTP server connection settings.

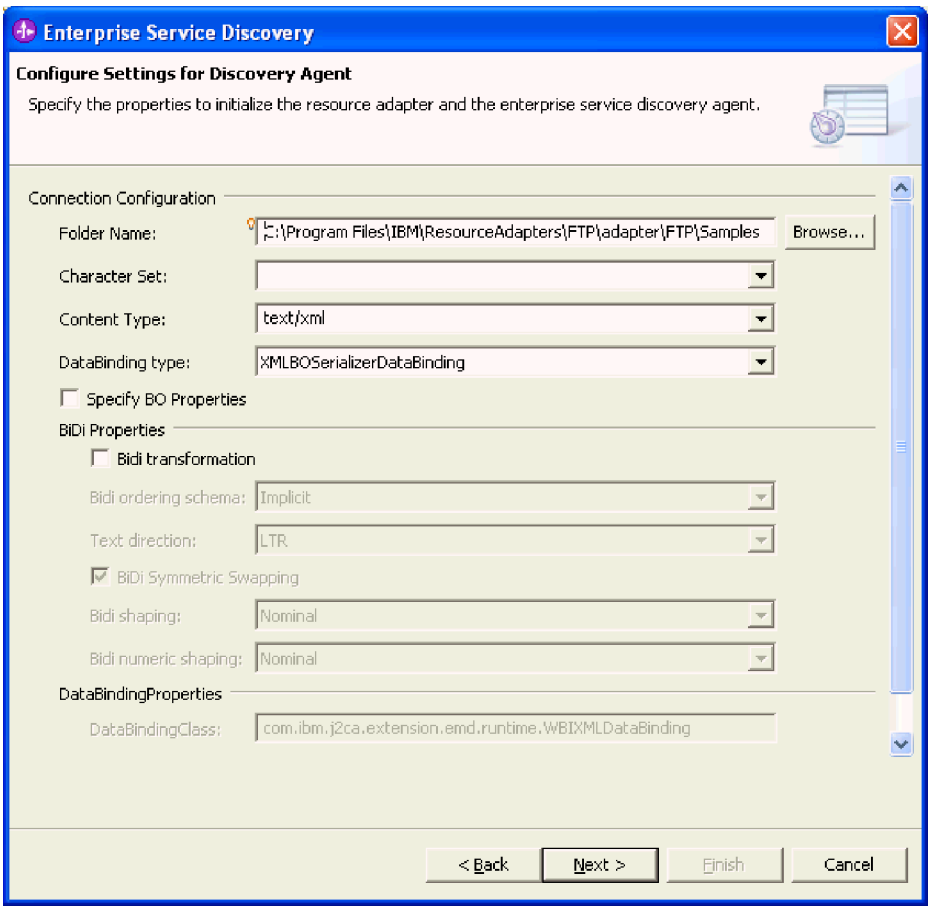
1. Switch to the Business Integration Perspective in WebSphere Integration Developer.
2. Right-click the frame of the Business Integration perspective window and select **New → Enterprise Service Discovery**. If **Enterprise Service Discovery** is not visible, select **Other** from the bottom of the menu. Then, in the Select a wizard window, expand the **Business Integration** folder, select **Enterprise Service Discovery**, and click **Next**.
3. Select **IBM WebSphere Adapter for FTP** from the **Import Configurations** menu and then click **Next**.



Select an Enterprise Service Resource Adapter window

4. In the Configure settings for Discovery Agent window, specify the properties that are used to discover the business data as well as for selecting the data binding that is used at runtime.
   a. Enter the **Folder name** where your XSD schemas for the business objects are stored, for example: `C:\Program Files\IBM\ResourceAdapters\FTP\adapter\FTP\samples\referencefiles`. The business objects to be used in the integration scenarios are selected from schema definitions present in the folder.
   b. **Optional:** Select a **Character Set**. Select the Character Set if the business data is of a different encoding. The business data corresponds to the data present in the files on which the operations are performed.

      Note: If using the samples scenarios generated by WebSphere Integration Developer, you do not need to select a Character Set.
   c. Enter a value in the **Content type** field. This one-time setting is used to bundle a content-type with a corresponding data binding. This displays all the content types that the adapter supports.
   d. **Optional:** If you want to configure individual business object properties, select the **Specify BO Properties** check box. If selected, you will be presented with a series of windows where you can define individual business object properties.
5. **Optional:** To define the log file and log level, click the **Show Advanced** button.
6. When you have defined all properties, select **Next**.



Configure Settings for Discovery Agent window

Result

The discovery service uses these properties to prepare a metadata tree that will be used for object selection and navigation in the following steps.

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

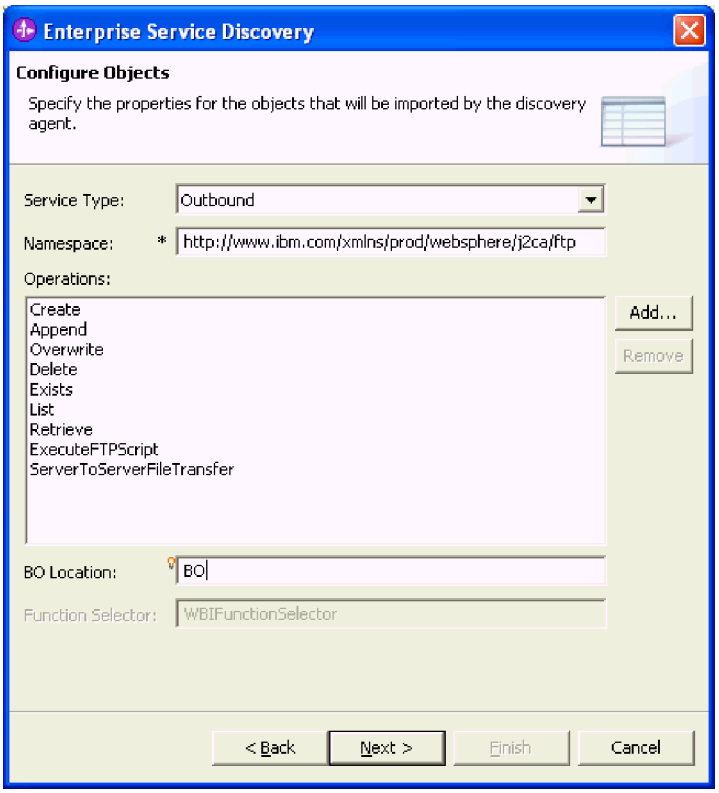
Select the business objects and services to be used by the adapter.

1. In the Find and Discover Enterprise Services window, select **Execute Query**.
2. Select the **Customer** object, then click **Add to import list**.
3. 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.



Configure objects window

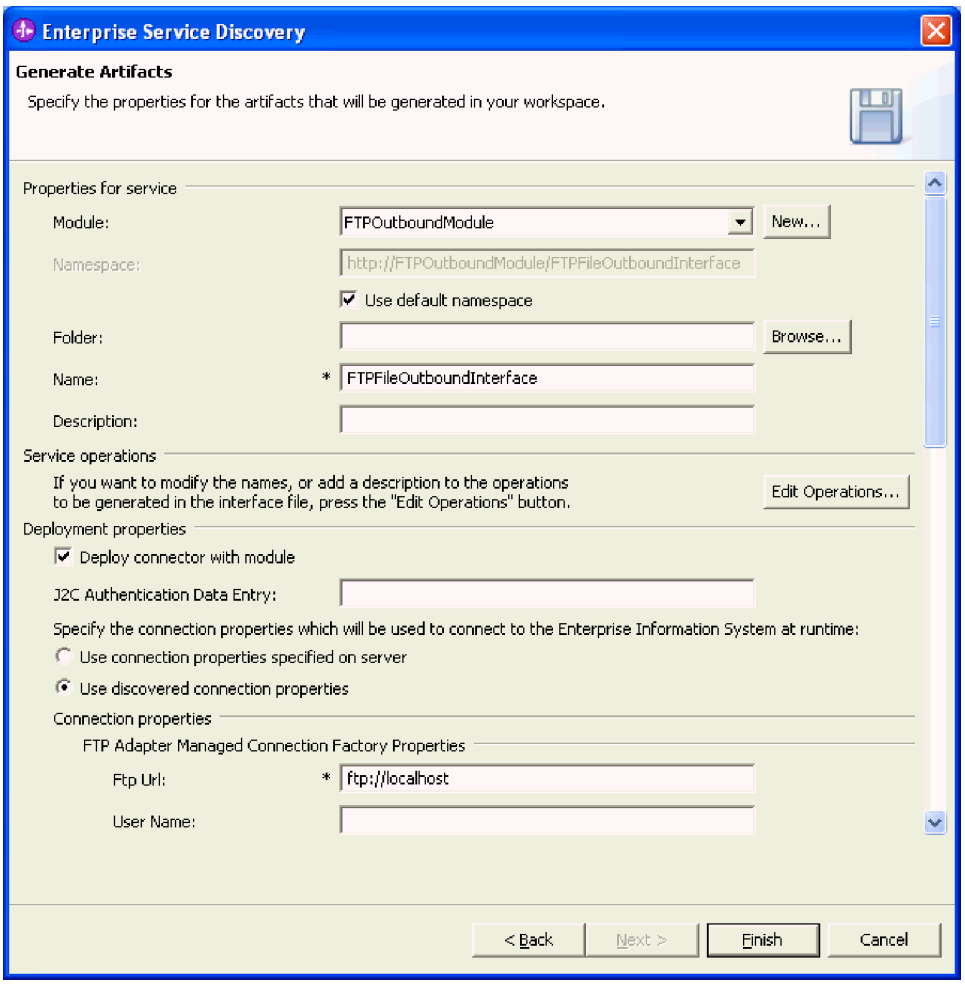
Result

The objects are now configured for outbound communication.

Generating artifacts

Generate business object definitions and related artifacts by using the enterprise service discovery wizard to first add a container business object to the business function and then create a new assembled adapter application, also referred to as an SCA module. After the business object definitions and related artifacts are generated, they are contained within the newly assembled adapter application (the SCA module).

1. In the Generate Artifacts window, click **New** next to the **Module** field to create a new module.
2. Select **Create a module project** and click **Next**.
3. In the New Module window, type a meaningful name in the **Module Name** field, then click **Finish**. The name you type is the name that will be given to the assembled adapter application (the SCA module) after the business object is generated.
4. In the Generate Artifacts window, select the **Use discovered connection properties** option.



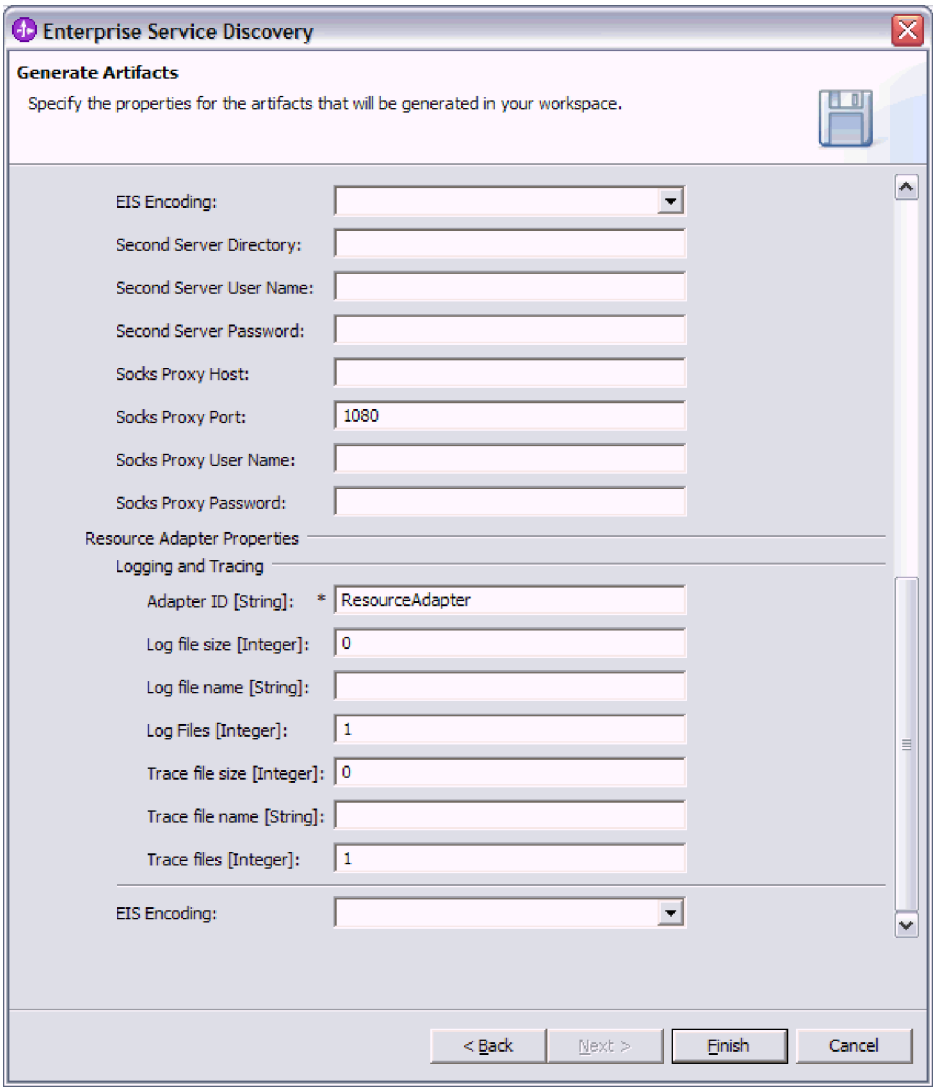
Generate Artifacts window

5. Enter the FTP Adapter Managed Connection Factory Properties. See "Managed (J2C) connection factory properties" on page 120 for more information on the properties. You must populate the **Ftp Url**, **User Name** and **Password** fields. The **Ftp Url** can be provided in one of the following formats:
   - `ftp://hostname[:portNumber]`
   - `ftp://[username:password@]hostname[:portNumber]`

     If using this format, you do not need to enter **User Name** and **Password** fields again.

     Note: The `portNumber` is optional in both of the above options.
   - `ftp://hostName`

     If using this format, provide the authentication alias by entering the value of the alias name in the **J2C Authentication Data Entry** property.
6. **Optional:** Scroll down to enter the logging and tracing properties.



Lower half of the Generate Artifacts window

7. Click **Finish**.

Result

The FTPFileOutboundInterface.wsdl and FTPFileOutboundInterface.import artifacts, and the FTPFileBG, FTPFile, 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.

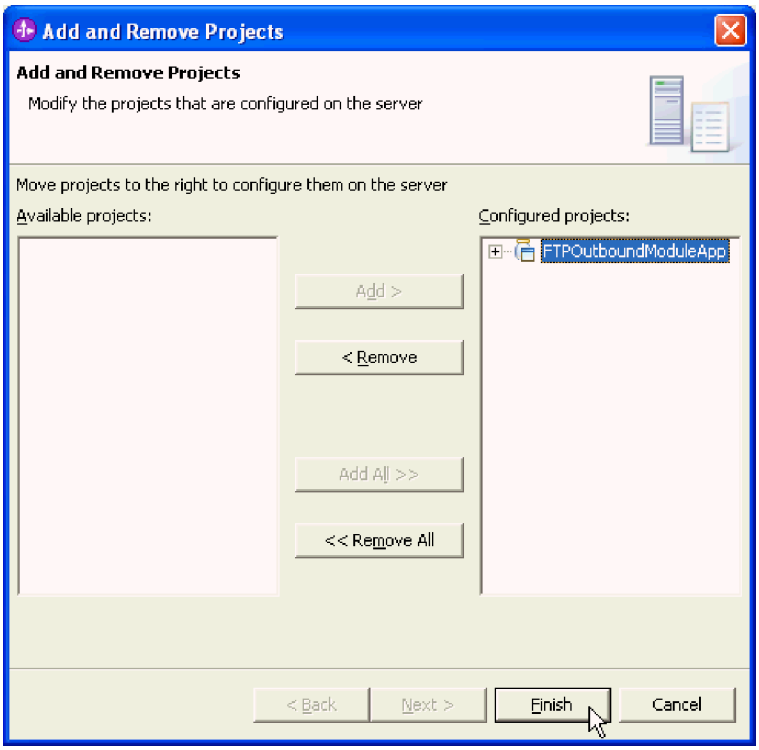
Deploying the module for testing

Deploy the Service Component Architecture (SCA) module in the WebSphere Integration Developer integration test client. The SCA module contains a server import or export.

1. Switch to the J2EE perspective by clicking **Window** → **Open perspective** → **Other**.
2. Add the SCA module to the server:
   a. Select the **Servers** tab.

b. Right-click the listed server and select **Add and remove projects**.
c. Select **FTPOutboundModuleApp** from the **Add and Remove Projects** window and click **Add**.

3. Click **Finish**.

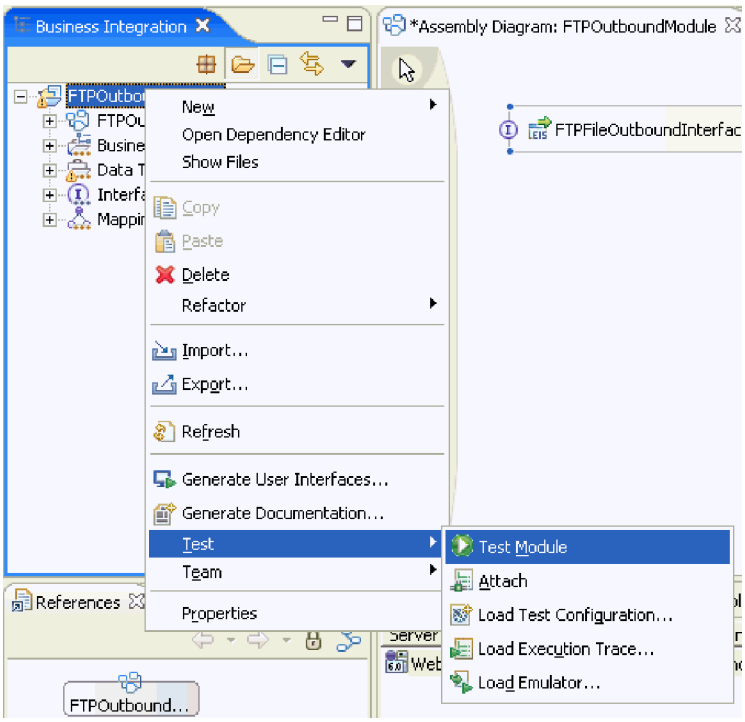


Add and Remove Projects window

Testing the module

Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Create a folder called outputdirectory in the user's home directory on the FTP server.
2. Launch WebSphere Integration Developer.
3. Switch to the Business Integration perspective.
4. Right-click the **FTPOutboundModule** project from the navigation panel and select **Test** → **Test Module**.



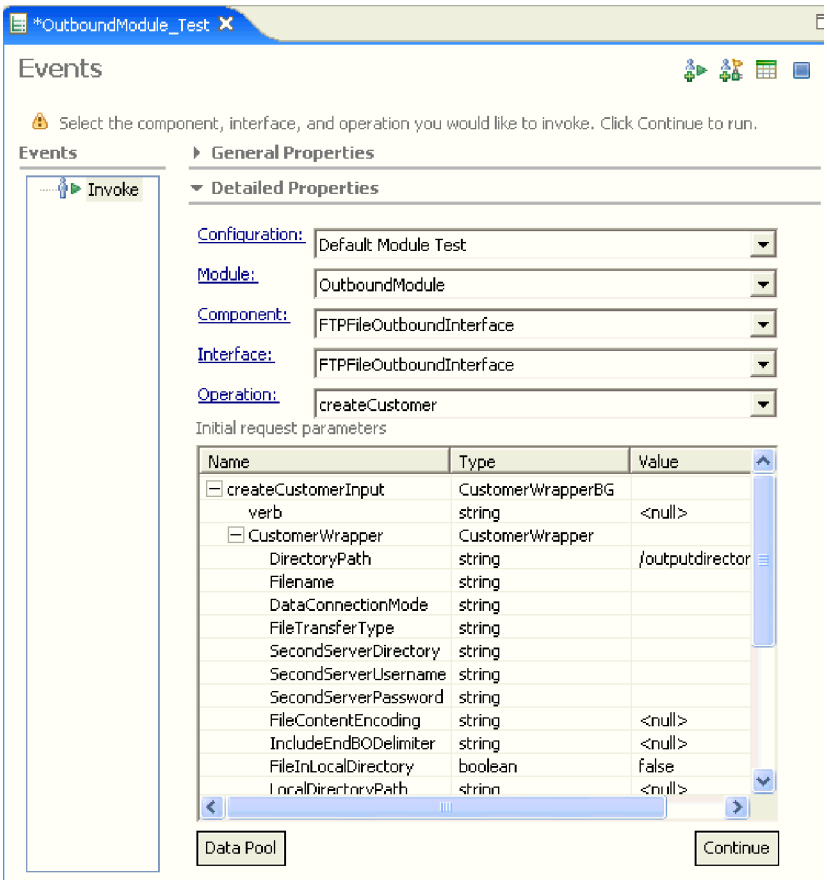
Test Module window

Testing tutorial 1: Outbound with data transformation for create operation

In this tutorial, you will configure the adapter to perform a create operation with data transformation. You will also see how to confirm the results of the operation.

1. In the Events window, select **createCustomer** from the **Operation** list.
2. Populate the properties values for the business object.

Name	Value
DirectoryPath	`/outputdirectory`
Filename	`Customer.bo`
CustomerName	*`Any name`*
Address	*`Any address`*
City	*`Any City`*
State	*`Any State`*



Events page with the createCustomer operation selected

3. Confirm that the Customer.bo file does not exist yet in the outputdirectory folder of the user's home directory on the FTP server.
4. Run the service by clicking **Continue**.
5. Confirm that the Customer.bo file exists in the outputdirectory folder.

Result

The Customer.bo file is created in the outputdirectory folder.

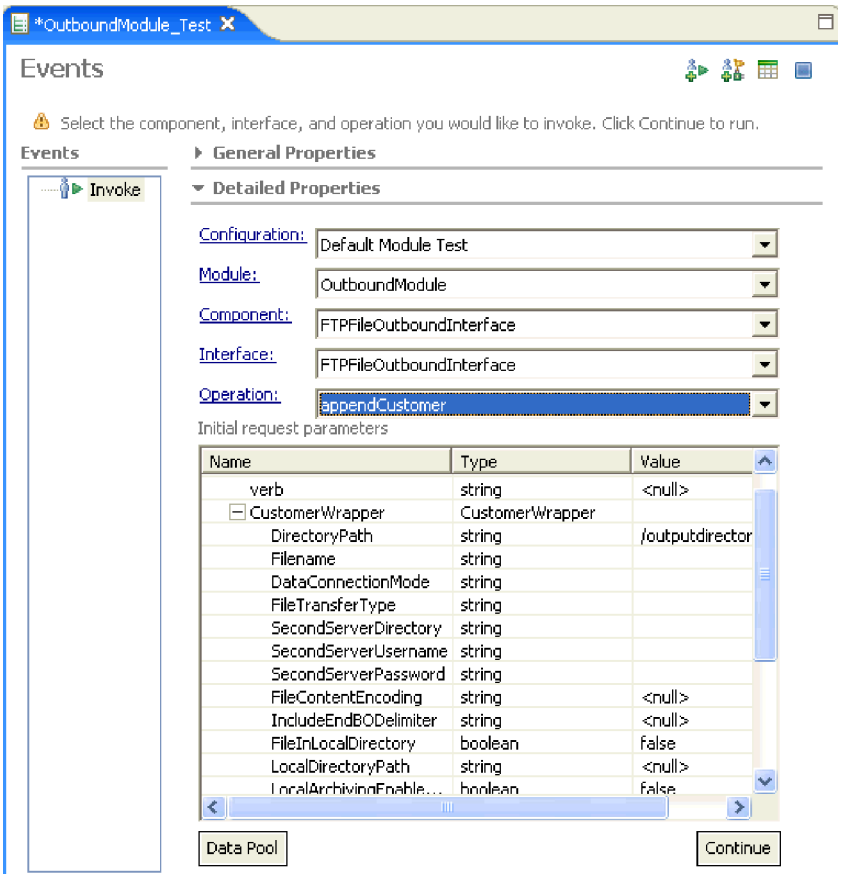
Testing tutorial 2: Outbound with data transformation for append operation

In this tutorial, you will configure the adapter to perform an append operation with data transformation. You will also see how to confirm the results of the operation.

1. In the Events window, click **Invoke**.
2. Select **appendCustomer** from the Operation list.
3. Populate the values for the business object.

Name	Value
DirectoryPath	`/outputdirectory`
Filename	`Customer.bo`
CustomerName	*`Any name`*
Address	*`Any address`*

Name	Value
City	*Any City*
State	*Any State*



Events page with the appendCustomer operation selected

4. Run the service by clicking **Continue**.
5. Check the output of the service, and the data in the outputdirectory folder to ensure that it matches the expected results.

Result

The existing Customer.bo business object will be appended to.

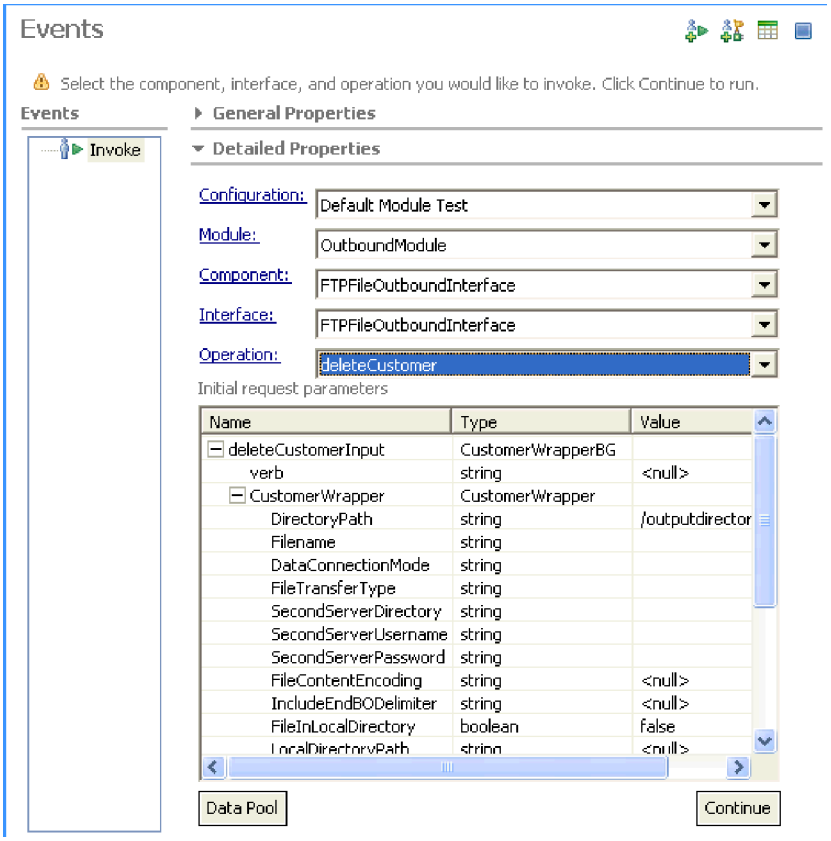
Testing tutorial 3: Outbound for delete operation

In this tutorial, you will configure the adapter to perform a delete operation. You will also see how to confirm the results of the operation.

1. In the Events window, click **Invoke**.
2. Select **deleteFTPFile** from the Operation list.
3. Populate the values for the business object.

Name	Value
DirectoryPath	`/outputdirectory`
Filename	`Customer.bo`
CustomerName	*Any name*

Name	Value
Address	*Any address*
City	*Any City*
State	*Any State*



Events page with the deleteCustomer operation selected

4. Run the service by clicking **Continue**.
5. Check the output of the service and the data in the outputdirectory folder to ensure that it matches the expected results.

Result

The existing Customer.bo business object is deleted.

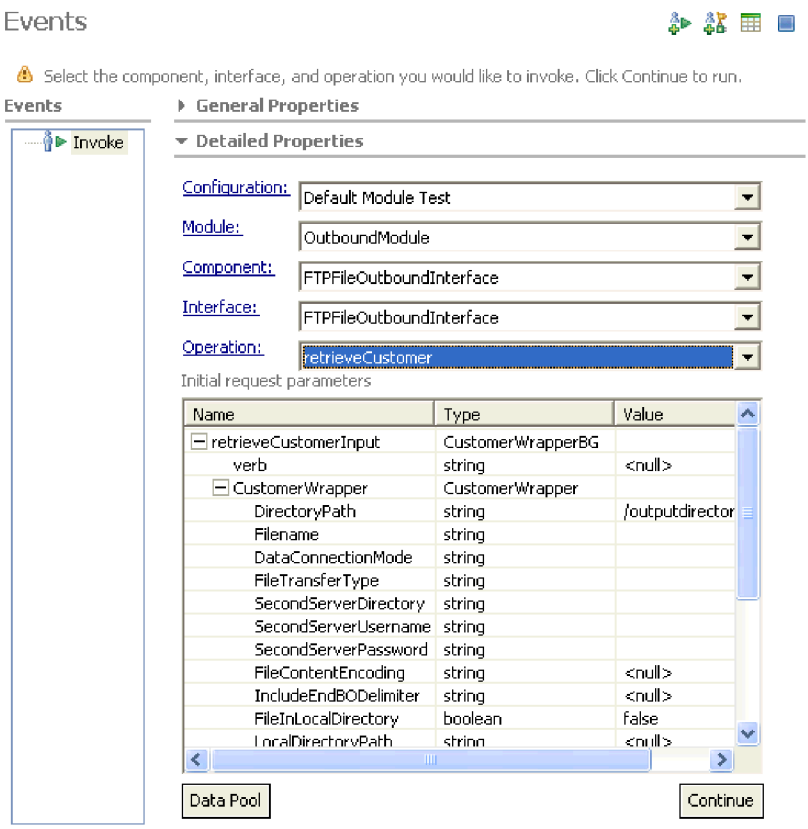
Testing tutorial 4: Outbound for retrieve operation

In this tutorial, you will configure the adapter to perform a retrieve operation. You will also see how to confirm the results of the operation.

1. In the Events window, click the **Invoke** button.
2. Select **retrieveFTPFile** from the Operation list.
3. Populate the values for the business object.

Name	Value
DirectoryPath	`/outputdirectory`
Filename	`Customer.bo`
CustomerName	*Any name*

Name	Value
Address	*Any address*
City	*Any City*
State	*Any State*



Events page with the retrieve customer operation selected

4. Run the service by clicking **Continue**.
5. Check the output of the service and the data in the outputdirectory folder to ensure that it matches the expected results.

Result

The content of the file which is specified in the request is returned.

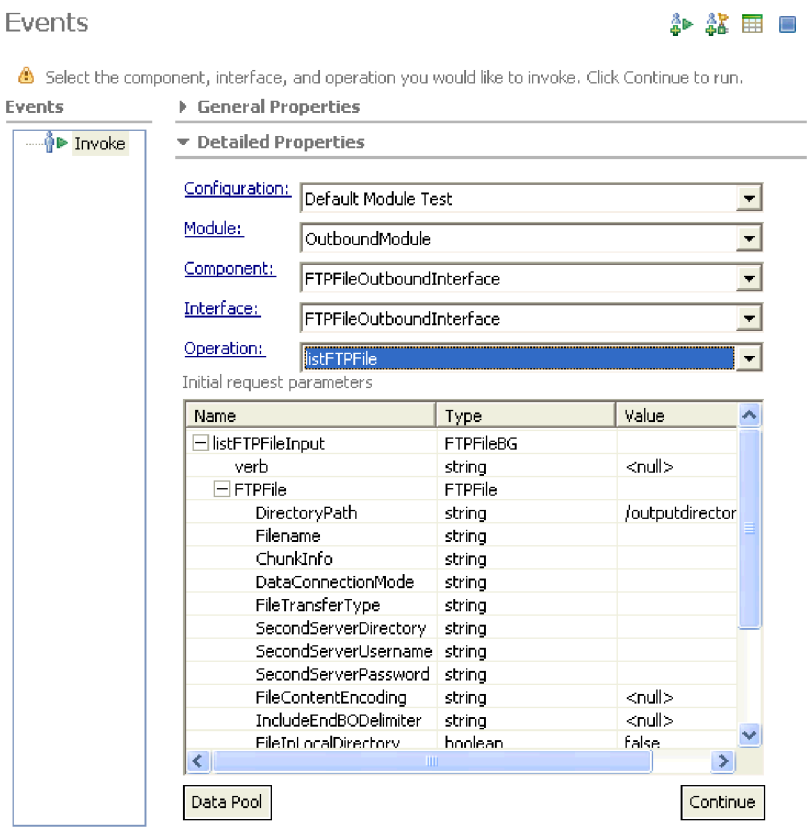
Testing tutorial 5: Outbound for list operation

In this tutorial, you will configure the adapter to perform a list operation. You will also see how to confirm the results of the operation.

1. In the Events window, click the **Invoke** button.
2. Select **listFTPFile** from the Operation list.
3. Populate the values for the business object.

Name	Value
DirectoryPath	`/outputdirectory`
Filename	`Customer.bo`
CustomerName	*Any name*

Name	Value
Address	*Any address*
City	*Any City*
State	*Any State*



Events page with the list file operation selected

4. Run the service by clicking **Continue**.
5. Check the output of the service, and the data in the outputdirectory folder to ensure that it matches the expected results.

Result

The file names and directories specified in the request are returned.

Clearing the tutorial content

After completing the tutorial, you might want to remove the content from WebSphere Integration Developer. You can do this by deleting the **FTPOutboundModuleApp** adapter project.

1. Switch to the J2EE perspective in WebSphere Integration Developer. To do this, select **Window** > **Open Perspective** > **Other**. In the Select Perspective screen, select **J2EE**, then click **OK**.
2. In the Project Explorer pane, expand the Connector Projects folder.
3. Right-click **FTPOutboundModuleApp**, then select **Delete**.
4. In the Delete Module Options screen, select **Also delete references to selected project(s)**, then click **OK**.

5. In the Confirm Project Delete screen, select **Also delete contents under '<*path_to_saved_project_data*>'**, then click **Yes**.

Tutorial: Receiving inbound data from the FTP server

The inbound tutorials demonstrate data transformation and pass through operations for data being received from the FTP server.

Creating the adapter project in WebSphere Integration Developer

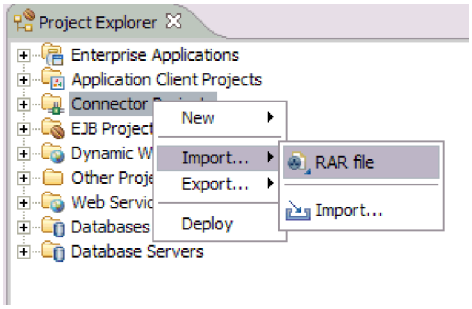
Use WebSphere Integration Developer to create a connector project and add the RAR file to the project.

About this task

Note: If you have already created the adapter project, you do not need to do it again.

How to perform this task

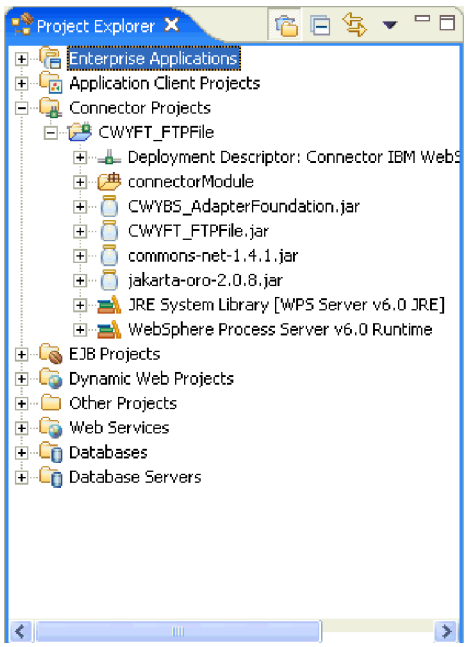
1. Click **Start** → **Programs** → **IBM WebSphere** → **Integration Developer 6.0** to Launch WebSphere Integration Developer.
2. Click **Window** → **Open perspective** → **Other** to switch to the J2EE perspective.
3. Right-click the Connector Projects folder, then select **Import** → **RAR file**.



4. In the Connector import window, click **Browse**, and select the RAR file location.
5. In the **Connector project** field, specify a project name.
6. Clear the **Add module to an EAR project** check box.
7. Accept all other defaults.
8. Click **Finish**.

Result

The RAR file is imported and a connector project is created in the workspace.



Configuring the adapter for inbound processing

Use the enterprise service discovery wizard to set the adapter configuration properties and to generate the necessary business objects to use for inbound processing.

Create the event database

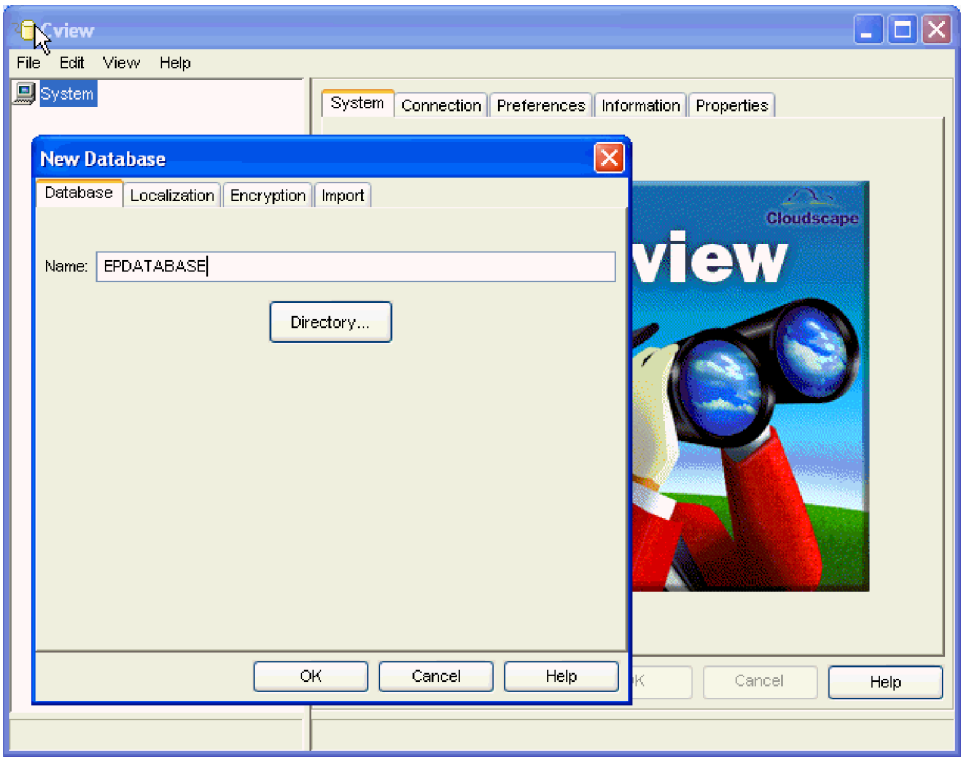
This scenario relies on having a database called EPDATABASE where the adapter will store events. After you create this database using the BAT file supplied in the samples folder, the FTP adapter will create the event table automatically.

About this task

Note: The server must not be running 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 `EPDATABASE`.
5. Select **OK** to close all windows.

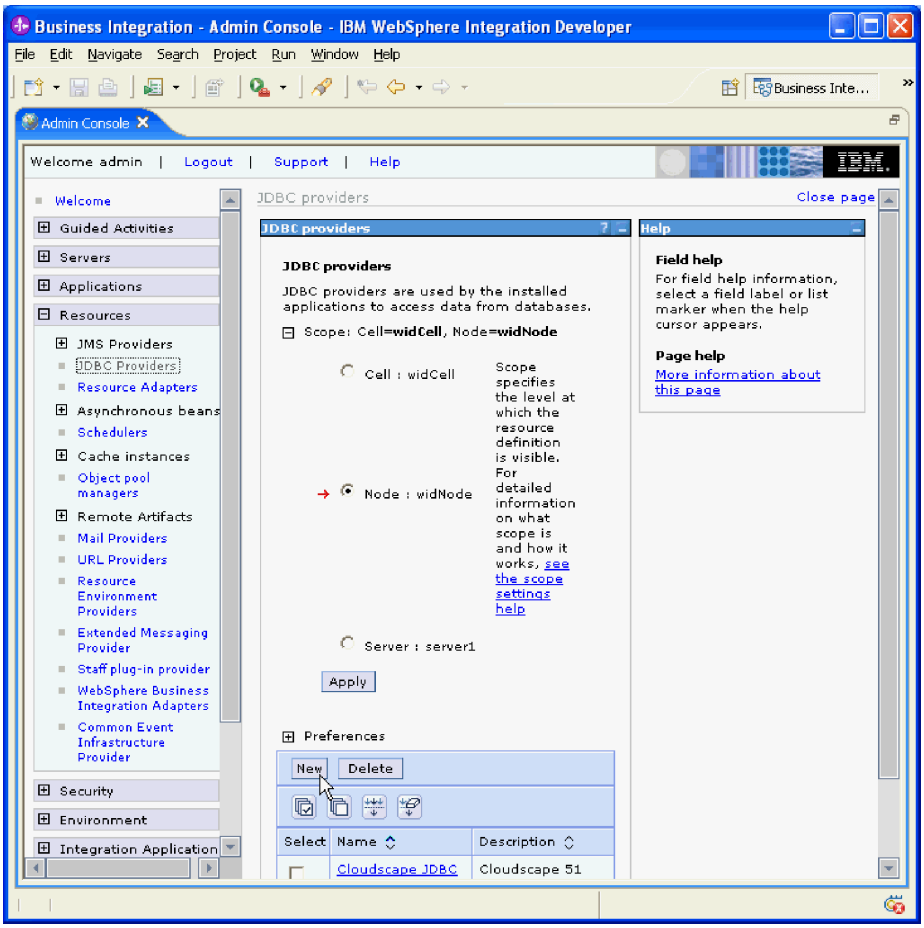


New database creation window

Configuring the data source

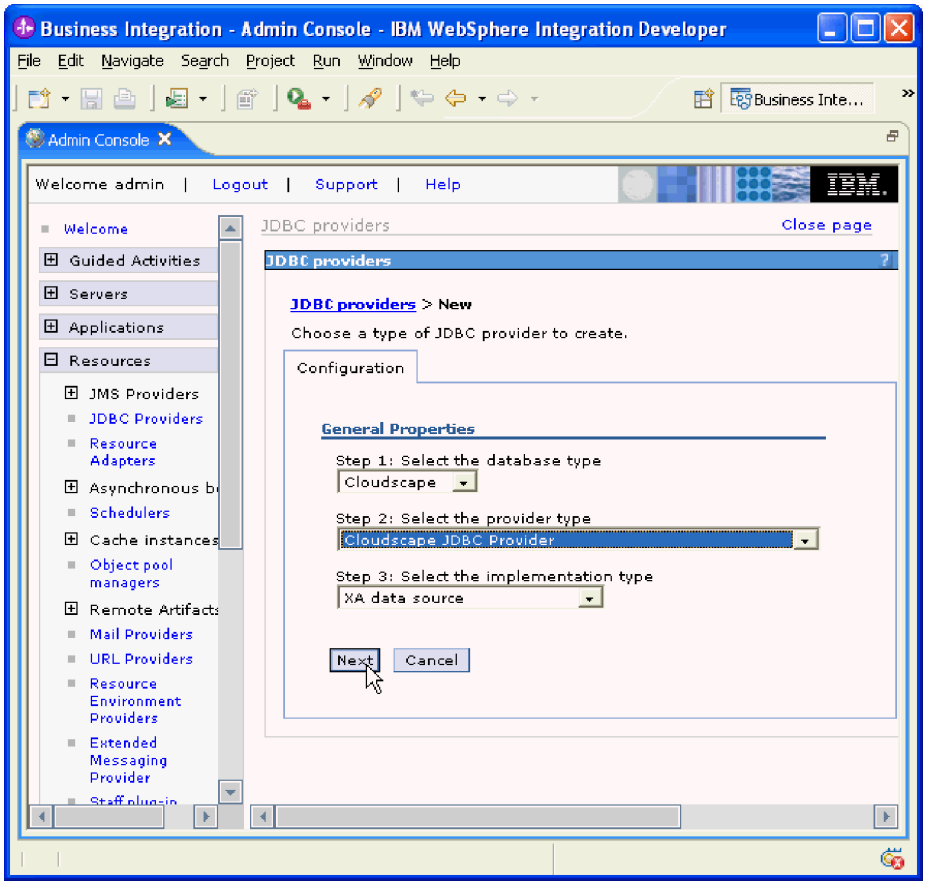
Configure WebSphere Integration developer to recognize the event database as the data source.

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 **WebSphere Process Server 6.0** and select **Start**.
4. Confirm that the server is active and ready for transaction requests by observing the status in the server view, located at the bottom of the Business Integration perspective. 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 EPDATABASE, 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**.



New JDBC providers pane

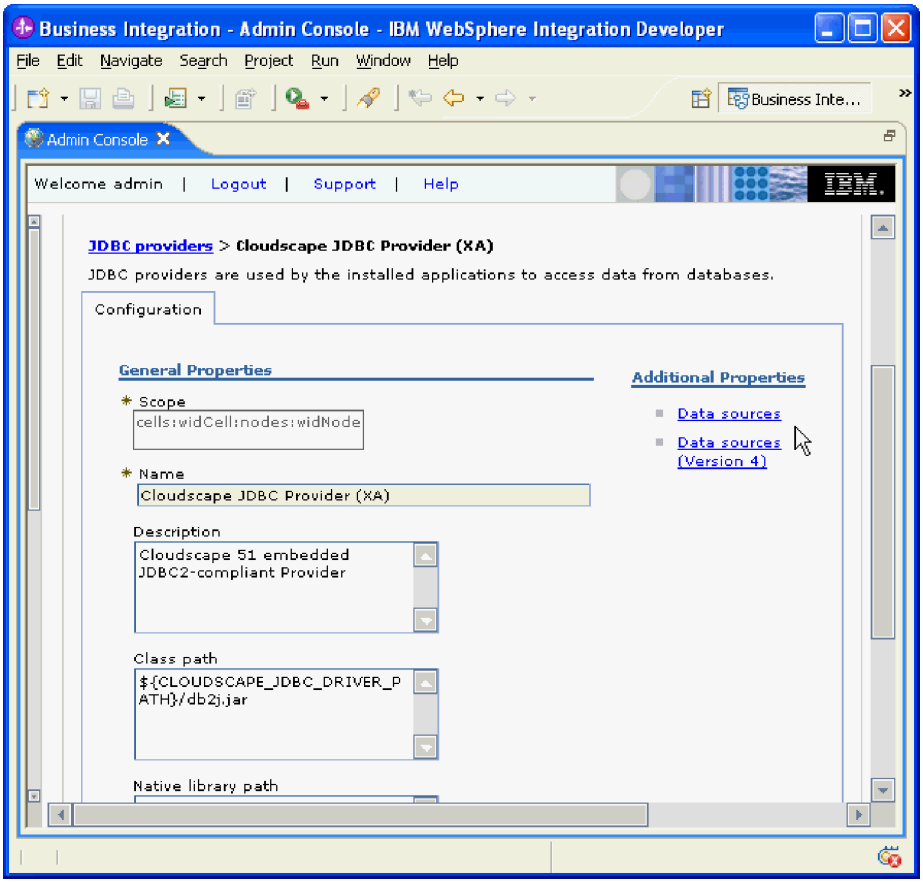
c. In the Configuration window, select **Cloudscape** as the database type, **Cloudscape JDBC Provider** as the provider type, and **XA data source** as the implementation type. Click **Next**.



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.



Data sources selection window

f. Select **New** and then type `EPJNDI` in the **JNDI name** field.

g. Scroll down to the **Database name** field and type `EPDATABASE`.

h. Click **OK**.

8. Click **Save** to apply the changes to the master configuration.
9. Return to the Data Sources window to select the data source you just created and to 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.`

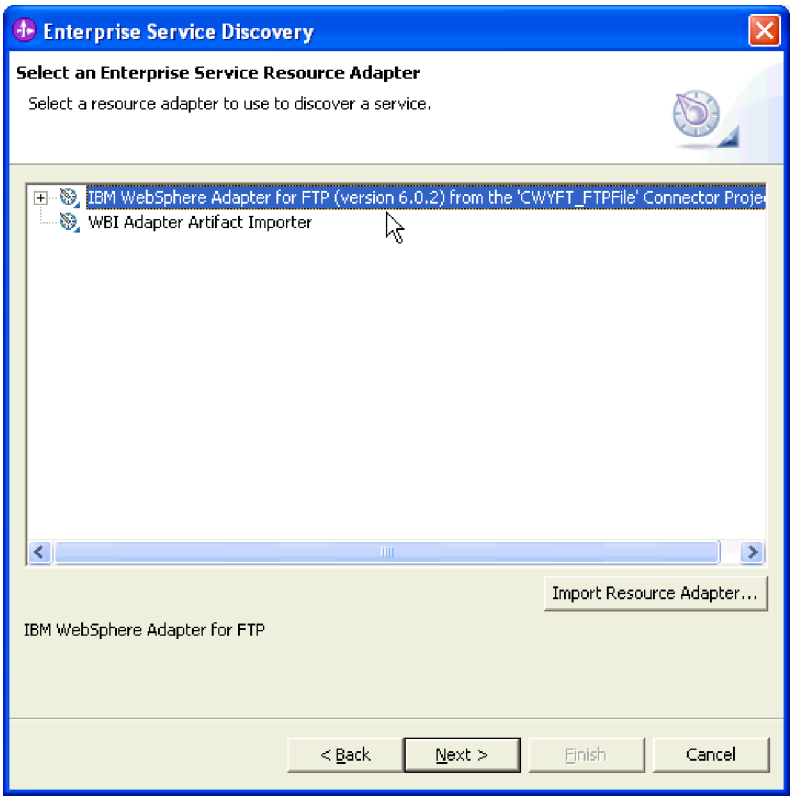
Note: The test connection must be successful to run the rest of the tutorial.

Setting connection properties for enterprise service discovery

Use enterprise service discovery to view all services available to the adapter and configure your FTP server connection settings.

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

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. Select **IBM WebSphere Adapter for FTP** from the **Import Configurations** menu and then click **Next**.

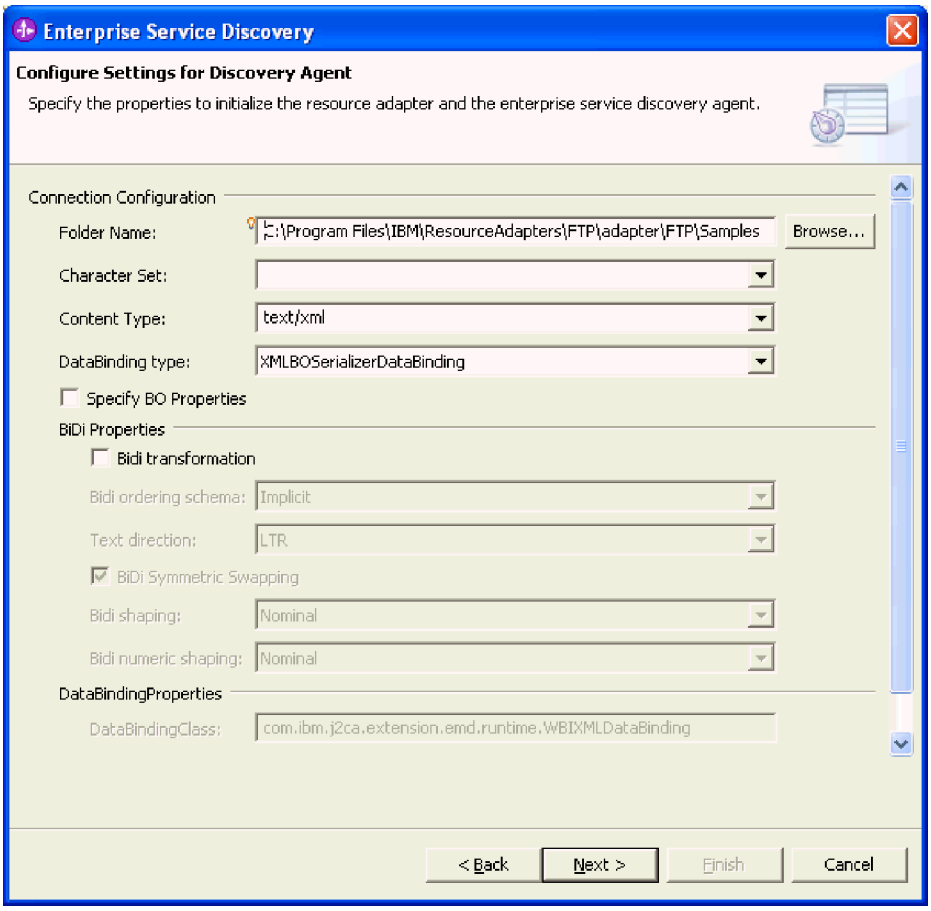


Enterprise Service Resource Adapter window

4. In the Configure settings for Discovery Agent window, specify the properties that are used to discover the business data as well as for selecting the data binding that is used at run time.
   a. In the **Folder name** field, type the name of the folder where your XSD schemas for the business objects are stored. For example: `C:\Program Files\IBM\ResourceAdapters\FTP\adapter\FTP\samples\referencefiles`. The business objects to be used in the integration scenarios are selected from schema definitions present in the folder.
   b. **Optional:** Select a **Character Set**. Select the Character Set if the business data is of a different encoding. The business data corresponds to the data present in the files on which the operations are performed.

      Note: If using the samples scenarios generated by WebSphere Integration Developer, you do not need to select a Character Set.
   c. Enter **Content type**. This one-time setting is used to bundle a content-type with a corresponding data binding. This displays all the content types that the adapter supports.
   d. **Optional:** If you want to configure individual business object properties, select the **Specify BO Properties** check box. If checked, a series of windows will appear where individual business object properties can be defined.

5. **Optional:** Click the **Show Advanced** button to define the log file and log level.
6. When you have defined all properties, select **Next**.



Configure Settings for Discovery Agent window

Result

Using these properties, the discovery service prepares a metadata tree that you will use to select objects and navigate in the following steps.

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

Select the business objects and services to be used by the adapter.

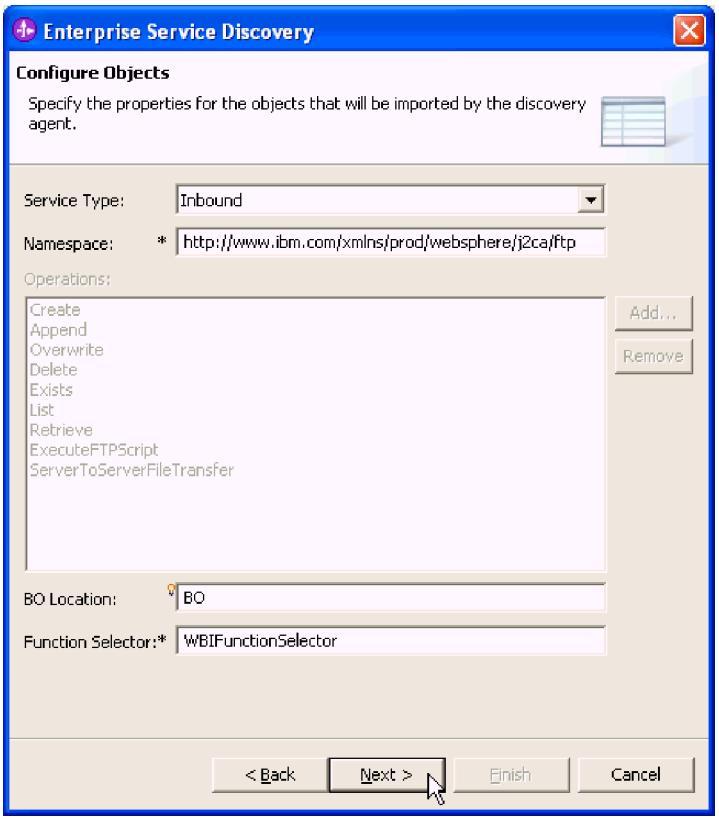
1. Select **Execute Query** in the Find and Discover Enterprise Services window.
2. Select the **Customer** object, then click **Add to import list**.
3. Click **Next**.

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 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.

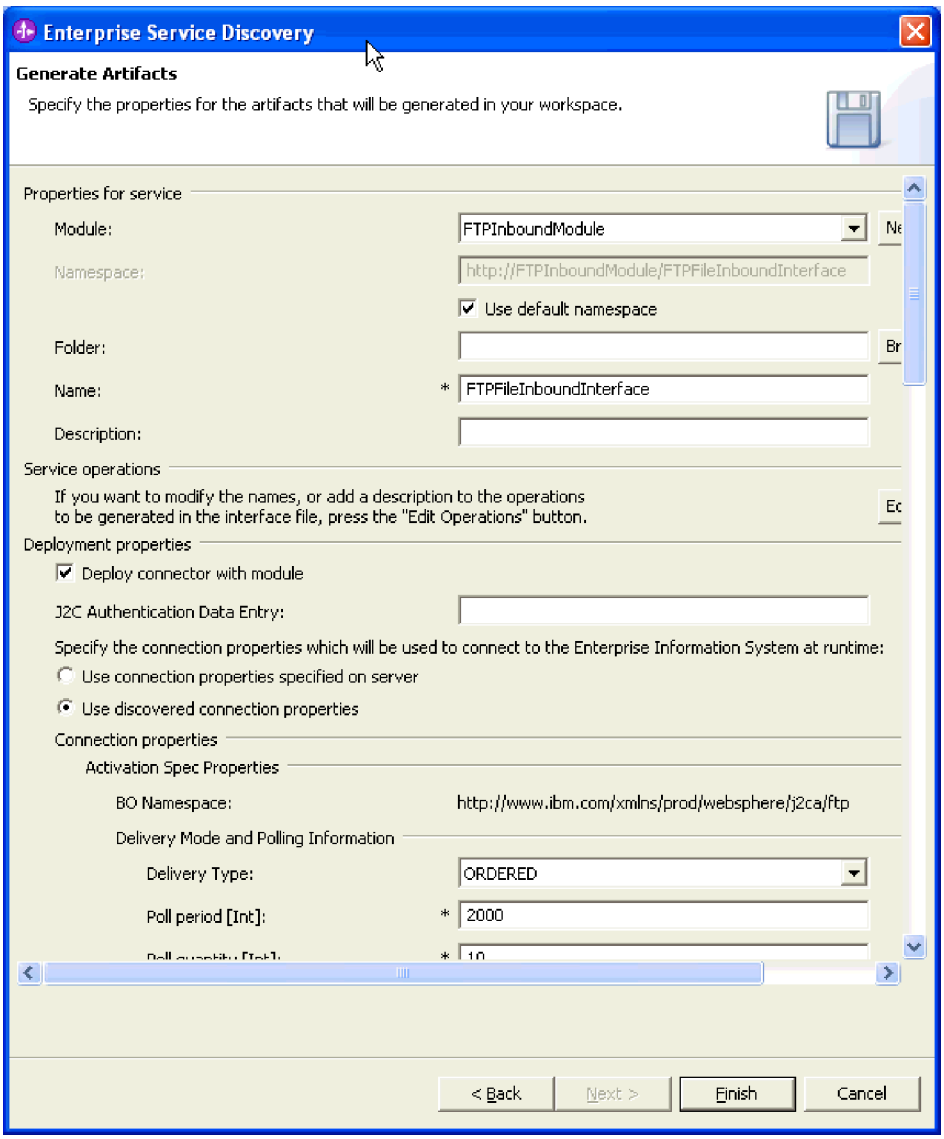


Configure objects window

Generating artifacts

Generate business object definitions and related artifacts by using the enterprise service discovery wizard to first add a container business object to the business function and then create a new assembled adapter application, also referred to as an SCA module. After the business object definitions and related artifacts are generated, they are contained within the newly assembled adapter application (the SCA module).

1. In the Generate Artifacts window, click **New** next to the **Module** field to create a new module.
2. In the New Module window, type a meaningful name in the **Module Name** field, then click **Finish**. The name you type is the name that will be given to the assembled adapter application (the SCA module) after the business object is generated.
3. In the Generate Artifacts window, select the **Use discovered connection properties** option.



Generate Artifacts window

4. Scroll down to enter the required Activation Spec Properties, as indicated by the asterisk (*). See "Activation specification properties" on page 124 for more information on the properties. These required properties are:

   - **DataSource JNDI Name**

     The JNDI name used to create a JDBC connection from a data source (`EPJNDI`) that will be used by the adapter to store the events in the event persistence database table. The data source must already be configured in WebSphere Process Server's administrative console.

   - **Event Table Name**

     Table name that is created to store the events.

   - **Event Directory**

     `ftp://[username:password@]hostName[:portNumber]/eventDirectory`

     If username and password are not specified here, they must be specified in the **User Name** and **Password** fields.

   - **Event File Mask**

     Polls only the files that match the file mask, which by default is `*.*`.

- **FTP Get Quantity**

  Number of files retrieved from the remote FTP URL with each remote poll.
- **FTP Poll Frequency**

  Determines the frequency that the adapter polls the FTP server. For example, if set to 6, the adapter will poll the event directory once every 6 standard polls.
- **Local Event Directory**

  Specifies the local system directory where the adapter will download the event files from the FTP server.

5. **Optional:** Scroll down to enter the logging and tracing properties.
6. Click **Finish**.

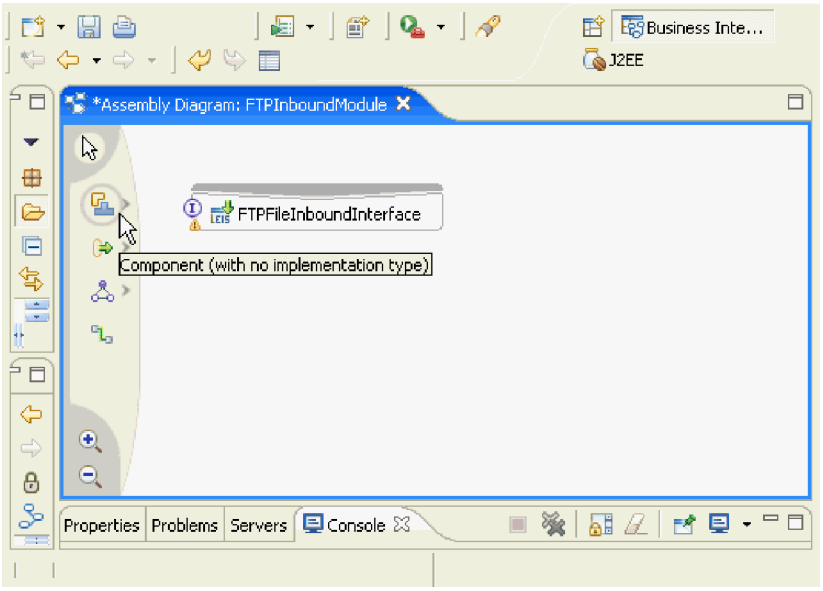
Result

The FTPFileInboundInterface.wsdl and FTPFileInboundInterface.export artifacts, and the FTPFileBG, FTPFile, 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

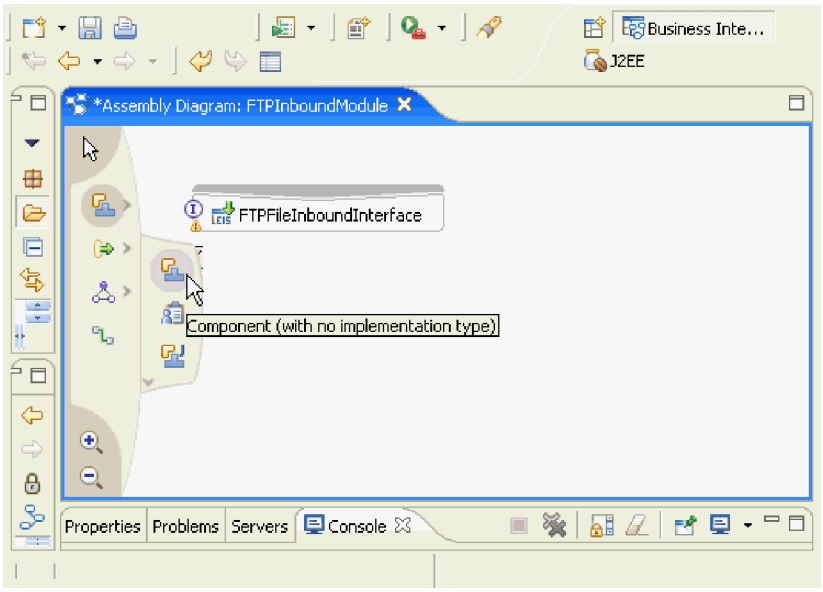
To link the adapter to other server processes create a reference binding to an adapter from the project module .

1. From the WebSphere Integration Developer window, switch to the Business Integration perspective.
   a. Select **Window** → **Open Perspective** → **Other**.
   b. From the list of perspectives that are displayed, select the **Business Integration**.
2. In the Business Integration perspective of WebSphere Integration Developer, right-click the module and click **Open With** → **Assembly Editor**.
   a. Click the **Component (with no implementation type)** icon.



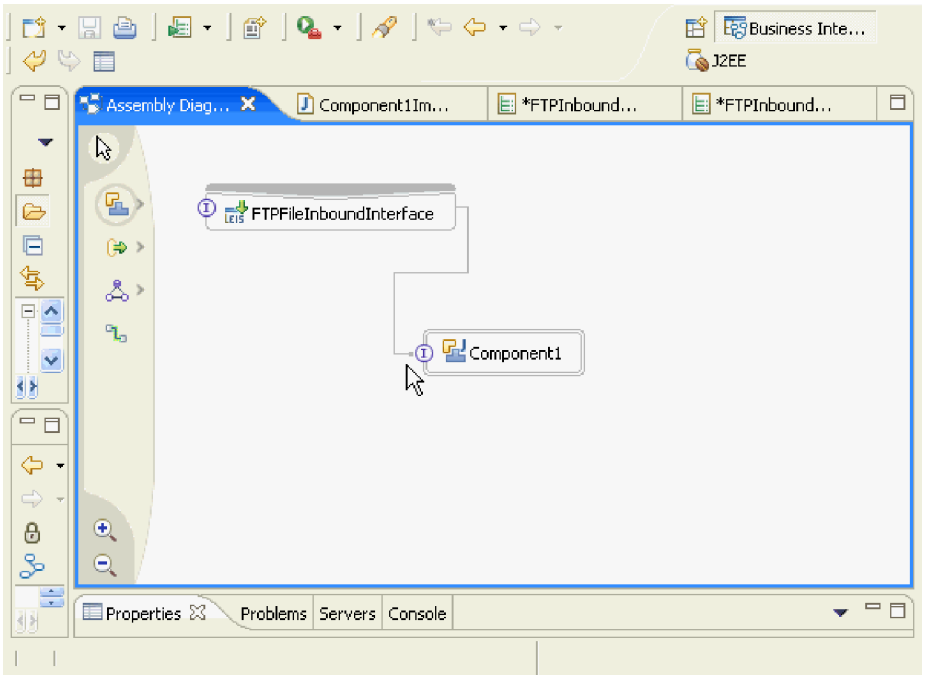
The Component icon in the Assembly Diagram window

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



Component icon

c. Click the **Wire** icon and drag the wire from **FTPInboundInterface** to **Component1** to create the wiring.



Component wiring window

d. Right-click **Component1** and select **Generate Implementation** → **Java**.

e. Select **(default package)** and click **OK**.

3. Click **File** → **Save**.

Result

The reference bindings are generated.

Deploying the module for testing

Deploy the Service Component Architecture (SCA) module in the WebSphere Integration Developer integration test client. The SCA module contains an EIS import or export.

1. Switch to the J2EE perspective by clicking **Window** → **Open perspective** → **Other**.
2. Add the SCA module to the server:
   a. Click the **Servers** tab.
   b. Right-click the listed server and select **Add and remove projects**.
   c. From the **Add and Remove Projects** window, select **FTPInboundModuleApp** and click **Add**.
3. Click **Finish**.

Testing the module

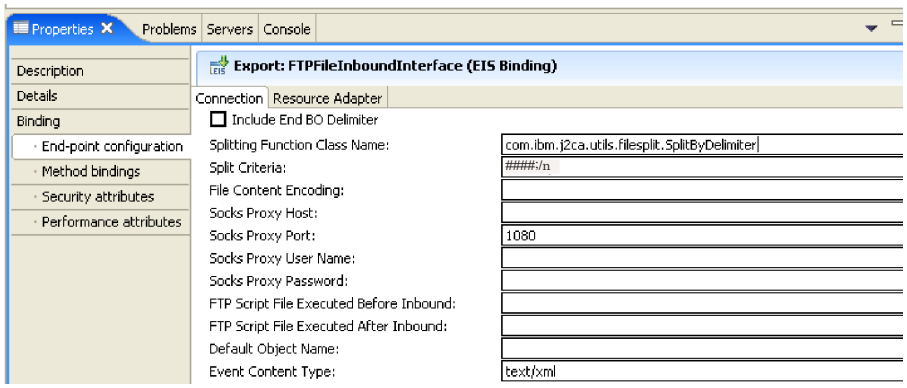
Test the assembled adapter application using the WebSphere Integration Developer integration test client.

1. Create a folder called eventdirectory in the user's home directory on the FTP server.
2. Launch WebSphere Integration Developer.
3. Switch to the Business Integration perspective.

Testing tutorial 6: Inbound with data transformation

In this tutorial, you will generate an inbound event that is processed using data transformation. You will also see how to confirm the results of the operation.

1. In the Business Integration perspective of WebSphere Integration Developer, right-click the **FTPInboundModule** module and click **Open With** → **Assembly Editor**.
2. Select the **Properties** tab and click **Binding** → **End-point configuration**.
3. Select **Connection** and expand **Activation Spec Properties**.
4. Specify the **Event Content Type**. For example: `text/xml`.
5. Configure file splitting based on a delimiter:
   a. Specify **Splitting Function Class Name** as `com.ibm.j2ca.utils.filesplit.SplitByDelimiter`.
   b. Specify **Split Criteria** as `####;\n`.



Connection configuration window showing Split Criteria

6. Click **File** → **Save**.

7. Right-click the **FTPInboundModule** project from the navigation panel and select **Test → Attach**. This attaches the integration test client to the EIS export.
8. Create a *.txt file in the eventdirectory, containing customer data, to trigger an inbound event. For example:

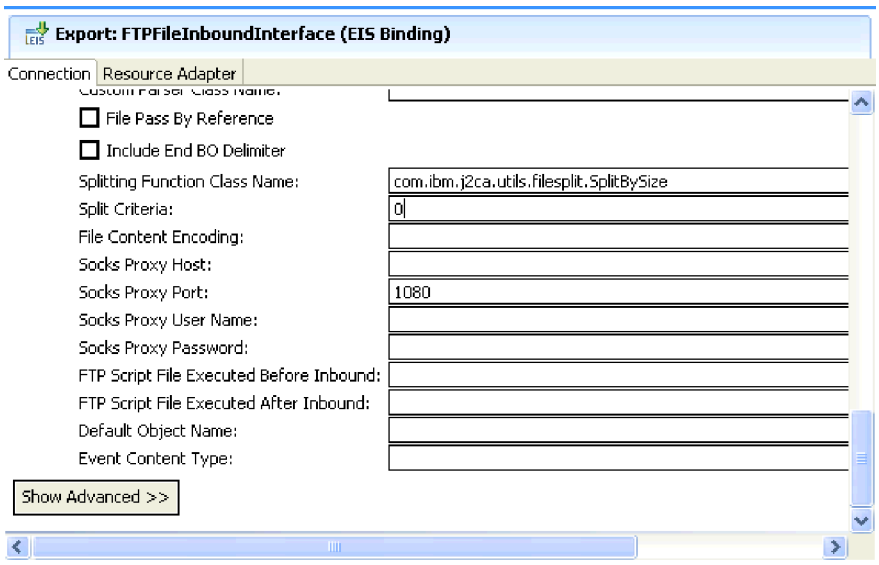
```
<?xml version="1.0" encoding="UTF-8"?>
<customer:Customer xsi:type="customer:Customer" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xmlns:customer="http://www.ibm.com/xmlns/prod/
websphere/j2ca/ftp/customer">
  <CustomerName>Michele</CustomerName>
  <Address>Da Vinci Ave.</Address>
  <City>NewYork</City>
  <State>NY</State>
</customer:Customer>
####
<?xml version="1.0" encoding="UTF-8"?>
<customer:Customer xsi:type="customer:Customer" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance"xmlns:customer="http://www.ibm.com/xmlns/prod/
websphere/j2ca/ftp/customer">
  <CustomerName>John</CustomerName>
  <Address>Grade A Colony</Address>
  <City>California</City>
  <State>CA</State>
</customer:Customer>
####
```

9. Click **Continue** to run the service and test the inbound module.
10. In the Select Deployment Location window, select **WebSphere Process Server v6.0** and click **Finish**.
11. Verify in the log and trace files that two business objects were posted to the end point. Log and trace files are located in the *install_root*/profiles/*profile_name*/logs/*server_name* folder.

Testing tutorial 7: Inbound with simple pass through

In this tutorial, you will generate an inbound event that is processed with simple passthrough. You will also see how to confirm the results of the operation.

1. In the Business Integration perspective of WebSphere Integration Developer, right-click the **FTPinboundModule** module and click **Open With → Assembly Editor**.
2. Select the **Properties** tab and click **Binding → End-point configuration**.
3. Select **Connection** and expand **Activation Spec Properties**.
4. Confirm that the **Event Content Type** contains no value.



Connection configuration window

5. Configure file splitting based on size:
   a. Specify **Splitting Function Class Name** as `com.ibm.j2ca.utils.filesplit.SplitBySize`.
   b. Specify **Split Criteria** as `0`.
6. Click **File** → **Save**.
7. Right-click the **FTPInboundModule** project from the navigation panel and select **Test** → **Attach**. This attaches the integration test client to the EIS export.
8. Create a *.txt file in the eventdirectory, containing customer data, to trigger an inbound event. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<customer:Customer xsi:type="customer:Customer" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance" xmlns:customer="http://www.ibm.com/xmlns/prod/
websphere/j2ca/ftp/customer">
  <CustomerName>Michele</CustomerName>
  <Address>Da Vinci Ave.</Address>
  <City>NewYork</City>
  <State>NY</State>
</customer:Customer>
####
<?xml version="1.0" encoding="UTF-8"?>
<customer:Customer xsi:type="customer:Customer" xmlns:xsi="http://www.w3.org
/2001/XMLSchema-instance"xmlns:customer="http://www.ibm.com/xmlns/prod/
websphere/j2ca/ftp/customer">
  <CustomerName>John</CustomerName>
  <Address>Grade A Colony</Address>
  <City>California</City>
  <State>CA</State>
</customer:Customer>
####
```

9. Click **Continue** to run the service and test the inbound module.
10. In the Select Deployment Location window, select **WebSphere Process Server v6.0** and click **Finish**.
11. Verify in the log and trace files that a single business object is posted to end point. Log and trace files are located in the *install_root*/profiles/*profile_name*/logs/*server_name* folder. Since **Split Criteria** is set to `0` and **Splitting Function Class Name** is set to `ibm.j2ca.utils.filesplit.SplitBySize`, the event file content is treated as single business object.

Clearing the tutorial content

After completing the tutorial, you may want to remove the content from WebSphere Integration Developer. You can do this by deleting the **FTPInboundModuleApp** adapter project.

1. Make sure you are in the J2EE perspective in WebSphere Integration Developer. To do this, select **Window** > **Open Perspective** > **Other**. In the Select Perspective screen, select **J2EE**, then click **OK**.
2. In the Project Explorer pane, expand the Connector Projects folder.
3. Right-click **FTPInboundModuleApp**, then select **Delete**.
4. In the Delete Module Options screen, select **Also delete references to selected project(s)**, then click **OK**.
5. In the Confirm Project Delete screen, select **Also delete contents under '<*path_to_saved_project_data*>'**, then click **Yes**.

Custom data bindings

WebSphere Process Server 6.0.2 provides the following sample custom EIS data bindings: Delimited, FixedWidth, and NameValue. The detailed steps for configuring these data bindings are located in *WPS_install_directory*\samples\doc\CustomEISDataBinding\index.html. The source code for the data binding is also provided and can be modified as needed.

Troubleshooting the tutorial

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

1. Export the project to an EAR file:
   a. From the WebSphere Integration Developer window, switch to the J2EE perspective.
   b. Right-click the project and select **Export → EAR file**.
   c. In the EAR Export window, select the EAR project and browse to the absolute path, including the EAR file.
   d. Select all check boxes and click **Finish**.
2. Click the **Server** tab in the bottom right corner of the Business Integration perspective.
3. Right-click **WebSphere Process Server 6.0** and select **Start**. This will start testing the server.
4. Confirm that the server is active and ready for transaction requests by observing the status in the server view, located at the bottom of the Business Integration perspective. The Console screen will read, `Server <server profile> is open for business`.
5. Open the administrative console window by right-clicking the server profile on the Server tab and selecting **Run Administrative Console**.
6. In the **user ID** field, enter the default administrative user ID `admin` and click **Log in**.
7. On the WebSphere Process Server Admin Console's Welcome admin tab, browse to **Applications → Enterprise Applications**.
8. Install the application:
   a. In the Enterprise Applications panel, select **Install**.
   b. Browse to the FTPInboundModuleApp.ear or FTPOutboundModuleApp.ear file, and click **Next**.

c. On the next few installation screens, accept all default options. Click **Next** until you reach the installation summary.
d. Click **Finish**. The `Application installed successfully` message will display when the installation completes.

9. Click **Save to Master Configuration**.

Starting the application using the administrative console

Complete the troubleshooting process by starting the application through the administrative console.

1. In the Enterprise Applications window, select the check-box next to the installed application.
2. Click **Start**.
3. Confirm that the application has started successfully.

Chapter 12. Reference information

Reference information supports the tasks you want to complete. The information includes all configurable properties for the FTP adapter, 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 and bidirectional 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.

Enterprise service discovery connection properties for the Adapter for FTP

Property	Type	Description	Required	Default value
ID	String	ID for the adapter type.	Yes	FTP
Description	String	Adapter description.	Yes	IBM WebSphere Adapter for FTP
DisplayName	String	Adapter display name.	Yes	IBM WebSphere Adapter for FTP
Vendor	String	Name of the vendor providing the adapter.	Yes	IBM
Version	String	Adapter version.	Yes	6.0.2

Connection configuration properties for the Adapter for FTP

Property	Type	Description	Required	Default value
Folder Name	WBIFolder Property	Folder where xsd files are located.	No	None
Content Type	Editable List of Values (formats supported by existing data handlers)	Content type to be used for all business objects. For example: text/xml.	No	None
Character Set	String	If the Specify Individual BO Props property is set to false, this value is added to the annotation of content-specific business objects. For example, Customer.xsd. If the Specify BO Props property *is set to true, this value is added to the annotation of the content-specific business object.	No	None
Data Binding Type	Editable List of Values (XMLBOSerializer DataBinding)	Name of the data binding that corresponds to the content type.	No	None

Property	Type	Description	Required	Default value
Data Binding Properties	Property Group. The list of properties is dependent on the selection of the data binding type.	Property group for the selected data binding type.	No	Null
Specify BO Props	Boolean	Set to true if you want to specify properties for individual object selection.	No	False

Business object properties for the Adapter for FTP

Property	Type	Description	Required	Default value
Content Type	Editable list of values (formats supported by existing data handlers)	Content type to be used for all business objects. For example: text/xml.	No	Content type selected in first window
Character Set	String	If the Specify BO Properties property is set to false, this value is added to the annotation of the content-specific business objects. For example, Customer.xsd. If the Specify BO Properties property is set to true, the Character Set value present in the Connection parameters for business object name window is added to the annotation of the content-specific business object.	No	Content type selected in first window
Data Binding Type	Editable list of values (XMLBOSerializer DataBinding)	Name of the data binding that corresponds to the content type.	No	Data binding type selected in the first window
Data Binding Properties	Property group (The list of properties is dependent on the selection of specific data binding in the above field)	Property group for the selected data binding type.	No	None
Expose in Service Description	Boolean	Determines if the selected object has to be exposed in the service description as an input or output type. When set to true, the business object is exposed as a type, and a content-specific wrapper business object is generated.	No	true

Selection properties for the Adapter for FTP

Property	Type	Required	Default value
ServiceType	List (Inbound or Outbound)	Yes	Outbound
Operations (outbound only)	MultiSelect (List of operations)	Yes	All operations
NameSpace	String	Yes	Provided by individual EMDs

Property	Type	Required	Default value
BO Location	String	No	None
Function Selector (inbound only)	String	Yes	WBIFunction Selector always

Adapter configuration properties

WebSphere Adapter for FTP has several categories of configuration properties: resource adapter, connection specification, managed (J2C) connection factory, activation specification, and interaction specification properties. These properties are configured using the enterprise service discovery wizard before deployment or with the WebSphere Application Server 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 WebSphere Process Server administrative console.

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

Resource adapter properties for the Adapter for FTP

Property	Type	Description	Globalized	Default value	Required
EISEncoding	String	Encoding of the FTP Server. Sets the encoding for the control connection while communicating with the FTP Server.		None	No
AdapterID	String	Identification of a deployment instance of the adapter.	Yes		Yes
LogFilename	String	Full path of the log file.	Yes		Yes
LogNumberOfFiles	Integer	Number of log files. When a log file reaches its maximum size, the adapter starts using another log file. If no value is specified, the adapter sets the number of log files to 1.	Yes	1	No
LogFileMaxSize	Integer	Size of the log files in kilobytes. If no value is specified, the file does not have a maximum size.	Yes	0	No
TraceFilename	String	Full path of the trace file.	Yes		No
TraceNumberOfFiles	Integer	Number of trace files to use. When a trace file reaches its maximum size, the adapter starts using another trace file. If no value is specified, the adapter sets the number of trace files to 1.	Yes	1	No

Property	Type	Description	Globalized	Default value	Required
TraceFileMaxSize	Integer	Size of the trace files in kilobytes. If no values is specified, then the file will have no maximum size.	Yes	0	No
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

Connection specification properties

Enterprise service discovery connection properties include outbound and inbound connection properties required for performing metadata discovery and bidirectional 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.

Connection specification properties for the Adapter for FTP

Property	Type	Description	Required	Default value
Username	String	Name of the user with privileges to connect to the FTP server and perform FTP operations. You do not need to specify a value for this attribute if the Username is included in the URL specified in the FtpUrl attribute.	No	None
Password	String	Password of the user with privileges to connect to the FTP server and perform FTP operations. You do not need to specify a value for this attribute if the password is included in the URL specified in the FtpUrl attribute.	No	None

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 managed (J2C) connection factory properties listed below.

Managed (J2C) connection factory properties for the Adapter for FTP

Property	Type	Description	Globalized	Bidirectional transformation supported	Default value	Required
FtpUrl	String	URL of the FTP server to which the connection is established during an outbound operation. The syntax for specifying the URL is: ftp://[UserId:password @]FTPserver [:port] The following information can also be specified in the FtpUrl property: • username and password of a user with privileges to connect to the FTP server and perform FTP operations. If not specified in FtpUrl, they must be specified in the Username and Password properties. • The FTP port. If not specified in FtpUrl, the adapter uses the default FTP port.	Yes	Yes	None	Yes
Username	String	Name of the user with privileges to connect to the FTP server and perform FTP operations. You do not need to specify a value for this attribute if the Username is included in the URL specified in the FtpUrl property.	Yes	Yes	None	No
Password	String	Password of the user with privileges to connect to the FTP server and perform FTP operations. You do not need to specify a value for this attribute if the Password is included in the URL specified in the FtpUrl property.	Yes	Yes	None	No
Socks Proxy Host	String	Host name of the workstation that is used as a proxy server through which the adapter requests are routed to the FTP server.	Yes	Yes	None	No
Socks Proxy Port	String	Port number of the proxy server through which the adapter requests are routed to the FTP server.	No	No	None	No
Socks Proxy Username	String	User name used to authenticate the proxy server.	Yes	Yes	None	No
Socks Proxy Password	String	Password used to authenticate the proxy server.	Yes	Yes	None	No
CustomParser ClassName	String	Fully qualified class name of the custom parser that is used to parse the `ls –l` output. Only used when the `ls –l` output deviates from standard output.	No	No	None	No

Property	Type	Description	Globalized	Bidirectional transformation supported	Default value	Required
EISEncoding	String	Encoding of the FTP server. Use this value to set the encoding for the control connection to the FTP server. • When both EISEncoding at the adapter level and EISEncoding at the MCF level are not set (both are null), nothing is set on the control connection while communicating with the FTP server. • When EISEncoding at the adapter level is set and EISEncoding at the MCF level is not set, the value at adapter level is set on the control connection while communicating with the FTP server. This is helpful when using multiple MCF's and the same encoding is set. In this case, set the value at the adapter level so that all the connections will have the same encoding for the control connection. • When EISEncoding at the adapter level is not set and EISEncoding at the MCF level is set, the value at MCF level is set on the control connection while communicating with the FTP server. Since the value is at the MCF level, this is applicable for only that MCF. • When both EISEncoding at the adapter level and EISEncoding at the MCF level are set, the value at the MCF level takes precedence. Specify any Java-supported encoding set for this attribute.	No	No	None	No

Property	Type	Description	Globalized	Bidirectional transformation supported	Default value	Required
Staging Directory	String	During an outbound create operation, the file is first created into this directory, if specified. After creation, the file is moved to the directory specified in the DirectoryPath property. The staging directory is also used for Append and Overwrite operations where the specified file is copied to StagingDirectory (if present), then appended or overwritten with content and moved back to the original specified directory. If StagingDirectory is not present, the operation is executed in the actual required directory.	Yes	No	None	No
SecondServer Directory	String	URL of the second FTP server to which the ServerToServerFileTransfer outbound operation is performed. Syntax for specifying the FTP URL is: ftp://[UserId:password@] FTPserver[:port]Directory ForSecondServer The following information can also be specified: • The username and password of a user with privileges to connect to the second FTP server and perform FTP operations. If not specified here, must be specified in the SecondServerUsername and SecondServerPassword properties. • The FTP port. If not specified here, the adapter uses the default FTP port. • The remote event directory. If not specified here, the adapter transfers the file to the directory to which the connection is established in the FTP server.	Yes	No	None	No
SecondServer Username	String	User name of the second FTP server to which the file is transferred during a server to server file transfer outbound operation.	Yes	No	None	No
SecondServer Password	String	Password of the Second FTP server to which the file is transferred during a server to server file transfer outbound operation.	Yes	No	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 WebSphere Process Server administrative console.

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

Activation specification properties for the Adapter for FTP

Property	Type	Description	Globalized	Required	Default value
DataSourceJNDIName	String	JNDI name of the data source used by event persistence to get the JDBC database connection. The data source must be created in WebSphere Process Server. The database name specified while creating the data source must already exist.	Yes	Yes	None
EventTableName	String	Name of the table that is used by the adapter for event persistence. When using multiple activation specifications, this value must be unique for each. The same table name must not be used by other instances of same adapter or a different adapter. If the table does not exist in the database, the adapter will create the table.	Yes	Yes	None
DatabaseSchemaName	String	Schema name of the database used by event persistence.	Yes	No	None
DatabaseUsername	String	Username used by event persistence for retrieving the JDBC database connection from the data source.	Yes	No	None
DatabasePassword	String	Password used by event persistence for retrieving the JDBC database connection from the data source.	Yes	No	None
CreateTable	Boolean	When set to true, the event table and related indexes are created. For troubleshooting table creation errors, set this property to false. The table and indexes can then be created manually.	No	No	true

Property	Type	Description	Globalized	Required	Default value
EventDirectory	String	FTP URL of the FTP server from where the event files are retrieved for inbound operations. Syntax for specifying FTP URL is: ftp://[UserId:password@] FTPserver[:port] [RemoteEventDirectory] The following information can also be specified: • Username and password of a user with privileges to connect to the FTP server and perform FTP operations. If not specified in EventDirectory, must be specified in the Username and Password properties. • FTP port. If not specified in EventDirectory, the adapter uses the default FTP port. • Remote event directory. If not specified in EventDirectory, the adapter polls the event files from the directory that the connection has established to the FTP server.	Yes	Yes	None
EventFileMask	String	Filter for the event files. The file filter is a well-qualified expression consisting of alphanumeric characters and the * and ? wild cards.	Yes	Yes	*.*
SortEventFiles	String	Determines the sorting order of event files being polled. Supported values are: • Filename – sort ascending on file name • Timestamp – sort ascending on last modified timestamp • <Blank> - not sorted Event file ordering from which events need to be delivered is valid only if the activation specification DeliveryType property is set to `ORDERED`. file name sorting is provided based on the locale of the FTP server. The ICU4J package is used to track the locales and their corresponding rules.	No	No	<blank> (= not sorted)

Property	Type	Description	Globalized	Required	Default value
FTPArchiveDirectory	String	Relative path of the archive directory on the FTP server. The directory must already exist. There are several options for using this property to specify archiving: • Specifying a value for this property, but no value for the FTPRenameExt property causes the adapter to append a timestamp to the event file name and move it to the FTP server archive directory specified in this property. • Specifying a value for this property and the FTPRenameExt property causes the adapter to rename the processed event file name with a timestamp and the value specified in FTPRenameExt and moves it to the FTP server archive directory specified in this property. • Specifying no value either for this property or the FTPRenameExt property causes the adapter to delete the processed event file without archiving it. • Specifying no value for this property but specifying a value for the FTPRenameExt property causes the adapter to rename the processed event file, adding a timestamp and the value specified in FTPRenameExt.	Yes	No	None
FTPRenameExtension	String	File extension or suffix that the adapter uses to rename the remote FTP file after the connector has polled for it. Renaming the file prevents the connector from polling the same file in the next poll cycle. The adapter can be configured to rename the processed event file and move it to an archive directory.	Yes	No	None
UserName	String	Name of the user who has privileges to connect to the FTP server and perform FTP operations. You do not need to specify a value for this property if the Username is included in the URL specified in the EventDirectory property.	Yes	No	None
Password	String	Password of the user who has privileges to connect to the FTP server and perform FTP operations. You do not need to specify a value for this property if the password is included in the URL specified in the EventDirectory property.	Yes	No	None

Property	Type	Description	Globalized	Required	Default value
FTPGetQuantity	Integer	Determines the number of files retrieved from the remote FTP URL with each remote poll.	No	Yes	10
FTPPollFrequency	Integer	Determines how frequently the adapter polls the FTP server, measured in the number of standard poll cycles. For example, if PollPeriod is set to 10000, and FTPPollFrequency is set to 6, the adapter polls the LocalEventDirectory every 10 seconds and polls the remote EventDirectory every 60 seconds. The adapter performs FTP polling only if you specify a value for this property. If PollPeriod is 0, we consider it as 1 for calculation. If the calculation evaluates to 0, the adapter does not perform FTP polling.	No	Yes	5

Property	Type	Description	Globalized	Required	Default value
EISEncoding	String	Encoding of the FTP server. Use this value to set the encoding for the control connection to the FTP server. • When both EISEncoding at the adapter level and EISEncoding at the activation specification level are not set (both are null), nothing is set on the control connection while communicating with the FTP server. • When EISEncoding at the adapter level is set and EISEncoding at the activation specification level is not set, the value at adapter level is set on the control connection while communicating with the FTP server. This is helpful when using multiple activation specification's and the same encoding is set. In this case, set the value at the adapter level so that all the connections will have the same encoding for the control connection. • When EISEncoding at the adapter level is not set and EISEncoding at the activation specification level is set, the value at activation specification level is set on the control connection while communicating with the FTP server. Since the value is at the activation specification level, this is applicable for only that activation specification. • When both EISEncoding at the adapter level and EISEncoding at the activation specification level are set, the value at the activation specification level takes precedence. Specify any Java-supported encoding set for this attribute.	No	No	None
LocalEventDirectory	String	Local system directory into which the adapter downloads event files from the FTP site. You must specify a value for this property to enable the adapter to process events.	Yes	Yes	None
LocalArchiveDirectory	String	Absolute path of the local Archive directory. The directory must be valid and already exist.	Yes	No	None

Property	Type	Description	Globalized	Required	Default value
FailedArchiveExt	String	File extension used to archive business objects in the event file that are not successfully processed. This property is used only when LocalArchiveDirectory is valid and exists.	Yes	No	fail
OriginalArchiveExt	String	File extension used to archive the original event file. This preserves the entire event file for reference in case any of its business objects fail. This property is used only when LocalArchiveDirectory is valid and exists.	Yes	No	original
SuccessArchiveExt	String	File extension used to archive all of the successfully processed business objects. This property is used only when LocalArchiveDirectory is valid and exists.	Yes	No	success
IncludeEndBODelimiter	Boolean	When set to true, the delimiter is sent with the business object content for further processing. This property is valid only when splitting the event files based on a delimiter.	No	No	false
DataConnectionMode	String	Data connection mode used by the FTP server during file transfers. Accepts either active or passive settings.	No	No	active
FileTransferType	Integer	File transfer type used during inbound operations. Accepts either ASCII or binary.	No	No	binary
CustomParserClassName	String	Fully qualified class name of the custom parser which is used to parse the `ls –l` output. Used only when the `ls –l` output deviates from standard output.	No	No	None
FilePassByReference	Boolean	Specifies that the file content of the event file is not sent to the endpoint. If set to true, the file is appended with a timestamp and sent to the LocalArchiveDirectory. The timestamp prevents errors and overwrites to the file when another file with the same name is received. This property can be set to true only when the LocalArchiveDirectory property is set and the specified directory exists. This property is used only for PassThrough inbound operations. When enabled, the file is not split into chunks.	No	No	false

Property	Type	Description	Globalized	Required	Default value
SplittingFunctionClass Name	String	This value takes the fully qualified class name of the class file to be used to enable file splitting. Requires two values: • The com.ibm.j2ca.extension. utils.filesplit.SplitByDelimiter class that splits the event file based on delimiter. • The com.ibm.j2ca.extension. utils.filesplit.SplitBySize class that splits the event file based on the event file size. The delimiter or file size is provided in the SplitCriteria property. If the EventContentType property is set to null, it is automatically set to a class name that performs splitting based on file size.	No	No	com.ibm. j2ca. extension. utils. filesplit. SplitBySize
FileContentEncoding	String	Encoding used to read the event files based on the EndBODelimiter property and during string to byte[] conversions. If not specified, the adapter attempts to read without any specific encoding. You can specify any Java supported encoding set.	No	No	None
SplitCriteria	String	This property takes different values based on the value of the SplittingFunctionClassName property. • If the SplittingFunctionClassName property specifies that files are split based on a delimiter, then SplitCriteria contains the delimiter that separates the business objects in the event file. • If SplittingFunctionClassName is set to a value which does splitting based on size, then the SplitCriteria property contains a valid number that represents the size in bytes. – If the event file size is greater than this value, the adapter splits the file into chunks of this size and the chunks are posted. – If the event file size is less than this value, the entire event file is posted. When SplitCriteria=0, chunking is disabled. When FilePassByReference is enabled during inbound PassThrough, the event file is not split.	Yes	No	0
SocksProxyHost	String	Host name of the machine used as a proxy server through which the adapter requests are routed to the FTP server.	Yes	No	None

Property	Type	Description	Globalized	Required	Default value
SocksProxyPort	String	Port number of the proxy server through which the adapter requests are routed to the FTP server.	No	No	None
SocksProxyUserName	String	User name used to authenticate the proxy server.	Yes	No	None
SocksProxyPassword	String	Password used to authenticate the proxy server.	Yes	No	None
FTPScriptFileExecuted BeforeInbound	String	Absolute path of the script file from the local adapter machine that gets run prior to every inbound poll cycle.	Yes	No	None
FTPScriptFile ExecutedAfterInbound	String	Absolute path of the script file from the local adapter machine that gets run after every inbound poll cycle.	Yes	No	None
DefaultObjectName	String	Wrapper business object name used by the Data Transformation Framework before delivering the event to the endpoint. For example, FTPFileBG or CustomerWrapperBG.	No	No	None
EventContentType	String	Content/MIME type of the input event file that is sent to the Data Transformation Framework to invoke the correct data handler while converting the business object string to a business object. This value is set to `NULL` for PassThrough operations.	No	No	None

Interaction specification properties

Enterprise service discovery connection properties include outbound and inbound connection properties required for performing metadata discovery and bidirectional 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.

Interaction Specification properties for the Adapter for FTP

Property	Type	Description	Globalized	Required	Default value
DirectoryPath	String	Absolute path of the directory on the FTP server where the outbound operation needs to be performed for all operations except ExecuteFTPScript, or the directory path on the local adapter workstation for the ExecuteFTPScript operation only. The directory must already existing.	Yes	Yes	None
Filename	String	Name of the file in the directory provided by the DirectoryPath property on which the outbound operations are performed. This value is required for all outbound operations except LIST.	Yes	No	None

Property	Type	Description	Globalized	Required	Default value
DataConnection Mode	String	Data connection mode used by the FTP server during file transfers. Takes either active or passive. This value is used only when a file transfer is taking place. This property is not used when performing a server to server file transfer outbound operation.	No	No	active
FileTransferType	Integer	File transfer type used during outbound operations. Takes either ASCII or binary.	No	No	binary
SecondServer Directory	String	URL of the second FTP server to which the server to server file transfer outbound operation is performed. Syntax for specifying FTP URL in the SecondServerDirectory attribute is: ftp://[UserId:password@]FTPserver[:port] DirectoryForSecondServer The following information can also be specified: • Username and password of a user with privileges to connect to the second FTP server and perform FTP operations. If not specified here, it must be specified in the SecondServerUsername and SecondServerPassword.properties. • FTP port. If not specified in here, the adapter uses the default FTP port. • Remote event directory. If not specified in here, the adapter transfers the file to the directory to which the connection is established in the FTP server.	Yes	No	None
SecondServer Username	String	Username of the second FTP server to which the file is transferred during a server to server file transfer outbound operation.	Yes	No	None
SecondServer Password	String	Password of the second FTP server to which the file is transferred during a server to server file transfer outbound operation.	Yes	No	None
FileContent Encoding	String	Encoding used while writing to the file. If this property is not specified, the adapter tries to read without using any specific encoding. You can specify any Java supported encoding set.	No	No	None
IncludeEndBO Delimiter	String	File content is appended with this value. Used during the outbound create, append, and overwrite operations.	Yes	No	None

Property	Type	Description	Globalized	Required	Default value
FileInLocal Directory	boolean	During outbound create operations, if this property is set to true, the file content is not available in the business object. The file is retrieved from the local directory on the adapter workstation. During outbound retrieve operations, if this property is set to true, the file content is not sent to the J2EE application as part of the business object. The file is saved to the local directory of the adapter workstation.	No	No	false
LocalDirectoryPath	String	During outbound create operations, when FileInLocalDirectory property is set to true, the file content is not available in the business object. Instead the file is picked from this directory. During outbound retrieve operations, when FileInLocalDirectory property is set to true, the file content is not sent to the J2EE application as part of business object. The file is saved to this directory.	Yes	No	None
LocalArchiving EnabledForCreate	boolean	During outbound create operations, when the file content is coming as part of the business object from a J2EE application and this property is set to true, the file is saved to the local workstation in the LocalArchiveDirForCreate directory before performing the outbound operation.	No	No	false
LocalArchiveDir ForCreate	String	During outbound create operations, when the file content is coming as part of the business object from a J2EE application and LocalArchivingEnabledForCreate is set to true, the file is saved to the local workstation in this directory.	Yes	No	None
StagingDirectory	String	During outbound create operations, the file will first be created into this directory. When the file creation is complete, the file is copied to the directory specified in the DirectoryPath property. This staging directory is also used for Append and Overwrite operations where the specified file is copied to the StagingDirectory, if present. The appended or overwritten content is then moved back to the original specified directory. If StagingDirectory is not specified, the operation is run in the actual required directory.	Yes	No	None

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.

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.

- EIS BiDi Format
- Metadata BiDi Format
- Skip BiDi Transformation
- EIS BiDi Special Format
- User Name BiDi Format
- Skip BiDi Transformation for User Name
- Password BiDi Format
- Skip BiDi Transformation for Password
- FTP URL BiDi Format
- Skip BiDi Transformation for FTP URL
- FTP URL BiDi Special Format
- Staging Directory BiDi Format
- Skip BiDi Transformation for Staging Directory
- Staging Directory BiDi Special Format
- Second Server Directory BiDi Format
- Skip BiDi Transformation for Second Server Directory
- Second Server Directory BiDi Special Format
- Second Server User Name BiDi Format
- Skip BiDi Transformation for Second Server User Name
- Second Server Password BiDi Format
- Skip BiDi Transformation for Second Server Password

Activation specification properties

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

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

- EIS BiDi Special Format
- User Name BiDi Format
- Skip BiDi Transformation for User Name
- Password BiDi Format
- Skip BiDi Transformation for Password
- Locale Event Directory BiDi Format
- Skip BiDi Transformation for Locale Event Directory
- Locale Event Directory BiDi Special Format
- Event Directory BiDi Format
- Skip BiDi Transformation for Event Directory
- Event Directory BiDi Special Format
- Event File Mask BiDi Format
- Skip BiDi Transformation for Event File Mask
- Event File Mask BiDi Special Format
- Locale Archive Directory BiDi Format
- Skip BiDi Transformation for Locale Archive Directory
- Locale Archive Directory BiDi Special Format
- Archive Directory BiDi Format
- Skip BiDi Transformation for Archive Directory
- Archive Directory BiDi Special Format
- FTP Script File Executed Before Inbound BiDi Format
- Skip BiDi Transformation for FTP Script File Executed Before Inbound
- FTP Script File Executed Before Inbound BiDi Special Format
- FTP Script File Executed After Inbound BiDi Format
- Skip BiDi Transformation for FTP Script File Executed After Inbound
- FTP Script File Executed After Inbound BiDi Special Format
- Rename Extension BiDi Format
- Skip BiDi Transformation for Rename Extension
- 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
- Split Criteria BiDi Format
- Skip BiDi Transformation for Split Criteria
- Event Persistance BiDi Format
- Skip BiDi Transformation for Event Persistence
- Data Source JNDI Name BiDi Special Format
- Skip BiDi Transformation for Data Source JNDI Name

Enterprise service discovery wizard help

The enterprise service discovery wizard is a tool that you use to configure the adapter for sending and receiving data to and from the enterprise information system. You can sequentially view field information presented by the wizard in the window information that follows.

Configure Settings for Discovery Agent window

With the Configure Settings for Discovery Agent window of the enterprise service discovery wizard, you can view the fields that are required to configure the adapter.

Before you use the enterprise service discovery wizard, certain connection properties must be set. These properties are described in the following table.

Help for enterprise service discovery wizard Configure Settings for Discovery Agent window, Adapter for FTP

Property	Description	Default value
FolderFF	Folder where you save the import or export.	None, the import or export is saved in the root folder of your module.
Log file output location	Storage location of the log file for enterprise metadata discovery.	[workspace directory]\.metadata\[adapter name]MetadataDiscovery.log
Logging Level	Controls the amount of information that you receive in the log. By default, you see only errors. If you want to generate a full trace, set the value for Logging Level to FINEST.	SEVERE
Module	The service component architecture module where you save the import or export.	None.
Name	Name of the adapter import or export that is generated. For example, [adapter name][inbound or outbound]Interface.	None.
Use connection properties specified on server	Select the check box for this property 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.	As the default, the check box is selected for *Use connection properties specified*.
Use discovered connection properties	Select the check box for this property if you want to specify the runtime properties during the enterprise metadata discovery process and save the values in the import or export.	This is not selected as a default value.

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**.

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 FTP.

Additional information you might need

Table 2. 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**.

Chapter 13. Glossary

The glossary of terms for IBM WebSphere Adapters is included in the WebSphere Adapters, version 6.0.2 information center.

You can view it at the following link: WebSphere Adapters glossary.

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

T

U

W

Printed in USA