WebSphere® Adapters

IBM

Version 6 Release 2

WebSphere Adapter for IBM i User Guide
Version 6 Release 2

WebSphere® Adapters

Version 6 Release 2

WebSphere Adapter for IBM i User Guide
Version 6 Release 2

Note

Before using this information and the product it supports, read the information in “Notices” on page 123.

December 2008

This edition applies to version 6, release 2, modification 0 of IBM WebSphere Adapter for IBM i and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

© Copyright International Business Machines Corporation 2006, 2008.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. Overview of WebSphere Adapter for IBM i

With WebSphere Adapter for IBM i, you can create modules that include the exchange of information with an IBM i system. By using the adapter, a module can send requests to read from or write to an IBM i data queue or run report program generator (RPG) programs without the need for special coding.

The adapter enables two-way communication between a module running on a supported application server or integration broker and an IBM i system. Using the adapter, a module can send requests to get data from an IBM i data queue or put data on an IBM i data queue without the need of special code. The adapter supports communication with both keyed and simple data queues such as first-in first-out (FIFO) or last-in first-out (LIFO). FIFO is a queuing technique in which the next item to be retrieved is the item that has been in the queue for the longest time. LIFO is a queuing technique in which the next item to be retrieved is the item most recently placed on the queue. A keyed data queue is an IBM i data queue that contains individual pieces of data (messages) that are associated with a key value. Messages can be received from a keyed data queue by specifying the key value. A client application can also receive events from the data queue, for example, it can be notified that a specific data queue has been updated. To process events that result from changes to the data queue, the adapter delivers events to an application. Using event notification, updates to the data queue can be automatically propagated to other applications. By combining event processing by WebSphere Adapter for IBM i and another adapter, updates can be automatically propagated to enterprise applications such as Siebel Business Applications, PeopleSoft Enterprise, and Oracle E-Business Suite.

The adapter is imported and configured in a module that is created in your development environment and deployed to your application server or integration broker. After being configured, the adapter acts like a service provider in part of a service oriented architecture (SOA) implementation, providing operations to get data from an IBM i data queue or put data on a IBM i data queue. Client applications interact with the module instead of directly with the IBM i system, so authentication details (such as user name and password) that you provide when you set up a module are shielded from client applications and services outside of the module. The adapter exposes a service interface that hides the mechanics of how the data or operations are obtained or performed.

What is the benefit? The module you create is a reusable unit designed to complete a specific inbound or outbound service. Each module uses a consistent interface and standard business objects, so applications consuming the service do not have to understand the lower-level details of the IBM i system.

Hardware and software requirements

The hardware and software requirements for WebSphere Adapters are provided on the IBM Support Web site.

To view hardware and software requirements for WebSphere Adapters, see http://www.ibm.com/support/docview.wss?uid=swg27006249

Additional information

The following links provide additional information you might need to configure and deploy your adapter:

- The compatibility matrix for WebSphere Business Integration Adapters and WebSphere Adapters identifies the supported versions of required software for your adapter. To view this document, go to the WebSphere Adapters support page and click **Compatibility Matrix** beneath the **Related** heading in the **Additional support links** section: http://www.ibm.com/software/integration/wbiadapters/support/.
- Technotes for WebSphere Adapters provide workarounds and additional information that are not included in the product documentation. To view the technotes for your adapter, go to the following Web page, select the name of your adapter from the **Product category** list, and click the search icon: http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm.

Technical overview of WebSphere Adapter for IBM i

The adapter supports IBM i data queue integration with modules running on a supported application server or integration broker. The adapter provides outbound and inbound processing under the Java™ 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) and integrates with Service Component Architecture (SCA) components.

The adapter is housed within a module that you create in your development environment. Each module is designed to perform either an outbound or inbound service, such as adding a message to or retrieving a message from an IBM i data queue or running a report program generation (RPG) program on an IBM i system. The service encapsulated in the module provides a reusable unit that can be configured in your development environment and deployed on any supported application server.

Data queues on IBM i system have the following characteristics:

- The data queue allows for fast communication between jobs. Therefore, it is an excellent way to synchronize and pass data between jobs.
- Many jobs can simultaneously access the data queues.
- Messages on a data queue are free format.
- Fields are not required because they are in database files.
- The data queue can be used for either synchronous or asynchronous processing.
- The messages on a data queue can be ordered in one the following ways:
  - Last-in first-out (LIFO)
  - First-in first-out (FIFO)
  - Keyed

Each data queue object requires the integrated file system path name of the data queue. The system-recognized identifier is *DTAQ.

Outbound processing

Outbound processing refers to the operations performed when the module makes an outbound request to the adapter, such as adding messages to or removing messages from an IBM i data queue or running an RPG program on the IBM i system. The adapter provides a standard set of outbound operations.

Outbound processing begins when a module receives a business object from a service. This is called a request. When a request is received, the adapter uses the information stored in the business object and metadata from the interaction specification properties to perform an operation on the IBM i system. The adapter supports three outbound operations: GetQueue, PutQueue, and CallPGM.

Business objects represent data that the adapter needs to complete a service function, such as message content to be put on a data queue.

Figure 1 shows how the adapter and module function together as part of an outbound service. A module created for outbound processing receives the request, the adapter determines the operation to be performed based on the information received in the metadata from the interaction specification properties and then performs the operation on the IBM i system.

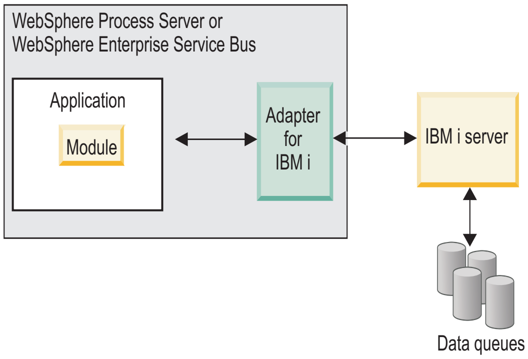


Figure 1. IBM i adapter as part of an outbound SOA implementation

Figure 2 on page 4 shows a more detailed view of the module in an outbound implementation. Each outbound module contains components that form a service, including an import. An import defines interactions between the Service Component Architecture (SCA) module and the service provider. An import has a binding and one or more interfaces. To communicate with references, imports require IBM i binding information which specifies how the data should be transported from the module. The references are bound to the physical location of the resource in the target operational environment at deployment. The WebSphere Integration Developer provides a graphical interface that you use to set up the import.

Interfaces and references are used by the component and import (see figure Figure 2 on page 4) to communicate with each other. Interfaces dictate to the users of a service, in this case the import, how the component can be used. It is a specification of the component's operations, which are GetQueue, PutQueue, or CallPGM for outbound requests. References declare what interface your service component will call. Each component in your module has one or more references. When you write your implementation for your service component using the WebSphere Integration Developer, you will call a reference instead of directly calling the component itself. This allows you to reassemble components in the future because you have not built dependencies into the code.

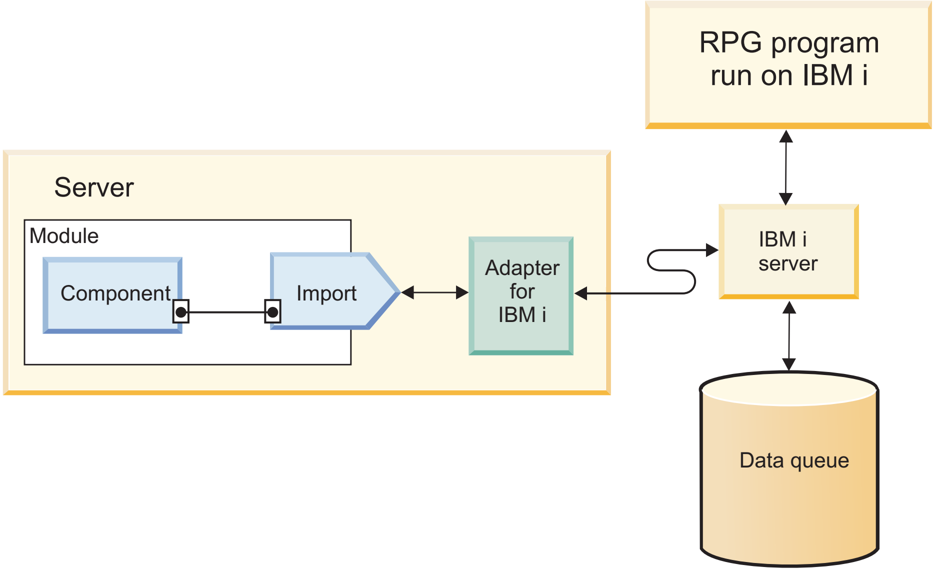


Figure 2. Outbound communication showing the import

Outbound operations

Components use operations to perform actions such as putting a message on an IBM i data queue, getting a message from an IBM i data queue, or running a RPG program on the IBM i system.

The adapter provides a standard set of outbound operations, specific to the type of business object being used. Table 1 lists the outbound operations that are supported for each type of business object.

Table 1. Outbound operations supported by type of business object

Business objects supported	Operation
Business object corresponding to a data queue	GetQueue
	PutQueue
Business object corresponding to an IBM i program	CallPGM

GetQueue operation:

The GetQueue operation retrieves a message from either first-in first-out (FIFO) or last-in first-out (LIFO) IBM i data queue or a keyed IBM i data queue.

To process the GetQueue operation for a FIFO or LIFO data queue, the adapter performs the following actions:

1. Receives a request in the form of a business object that needs to retrieve a message.
2. Fetches the message from the data queue by using the object path property in the interaction specification properties within the request. The object path is the absolute path of the data queue on the IBM i system.

   Note: If the IBM i data queue is classified as a FIFO data queue, the adapter fetches the first message on the queue. If the data queue is classified as a LIFO queue, the adapter fetches the last message on the queue.

3. Populates the request received in the form of a business object with the content of the message taken from the queue and returns it to the requesting component.

To process the GetQueue operation for keyed data queues, the adapter performs the following actions:

1. Receives a request in the form of a business object from a component that needs to retrieve a message from a keyed data queue. The message key attribute in the request contains the key value required to access the message on the IBM i system.
2. Fetches a message from the keyed data queue using the key value from the request and the object path property in the interaction specification properties. The object path is the absolute path of the data queue on the IBM i system.

   Note: Because keyed data queues do not support a LIFO storage, the adapter always fetches the first message from the keyed data queue.
3. Populates the business object with the content of the message corresponding to the key value taken from the keyed queue and returns it to the requesting component.

PutQueue operation:

The PutQueue operation puts a message on a first-in first-out (FIFO) or last-in first-out (LIFO) IBM i data queue.

When the PutQueue operation is performed by the adapter, a message is added to a data queue. Where the message is placed on the queue depends on the type of data queue such as FIFO or LIFO as defined on the IBM i system. The adapter supports two data queue types with the PutQueue operation: First-in first-out (FIFO) and Last-in first-out (LIFO). Messages added to a FIFO data queue are placed at the bottom of the queue and are the last in line to be processed by the IBM i system. Messages added to a LIFO data queue are placed at the top of the queue and are the first in line to be processed by the IBM i system.

To process the PutQueue operation, the adapter performs the following actions:

1. Receives a request in the form of a business object
2. Connects to the IBM i system
3. Converts information received from the business object into a message format
4. Adds the message to the IBM i data queue

CallPGM operation:

The CallPGM operation invokes an RPG program on an IBM i system.

CallPGM is used to call any PGM (RPG). RPG programs are stand-alone programs that can read or write data into data queues.

When the CallPGM operation is invoked, the adapter does the following:

1. Builds the parameter list for the RPG program from the attribute values received in the request business object
2. Establishes a connection with the IBM i system
3. Runs the program on the IBM i system

RPG programs do not return values. Instead, they take in all the input arguments provided by the adapter, complete the required processing, and write the result value into variables that match the same input arguments originally supplied. RPG programs are capable of returning business objects determined by the application-specific information in the passDirection type of parameter. The passDirection type can be Input, Output or InOut. If passDirection is classified as ″in,″ the adapter is an input. If passDirection is classified as ″out,″ the adapter is for output. If passDirection is classified as ″inout,″ then the adapter serves as both input and output.

Inbound processing

During inbound processing, the adapter polls an IBM i data queue at configurable intervals for new messages. The adapter removes the messages from the queue, places them in a staging queue, converts the messages into the business objects, and delivers the business objects to the client application.

Inbound processing enables a client application to receive notification when an object is put on an IBM i data queue. Figure 3 illustrates the flow of inbound processing. An IBM i application places data on a queue. Periodically, the adapter polls the queue, retrieves messages and processes events, and then delivers the events to a client application that runs in WebSphere Process Server or WebSphere Enterprise Service Bus or administrative console.

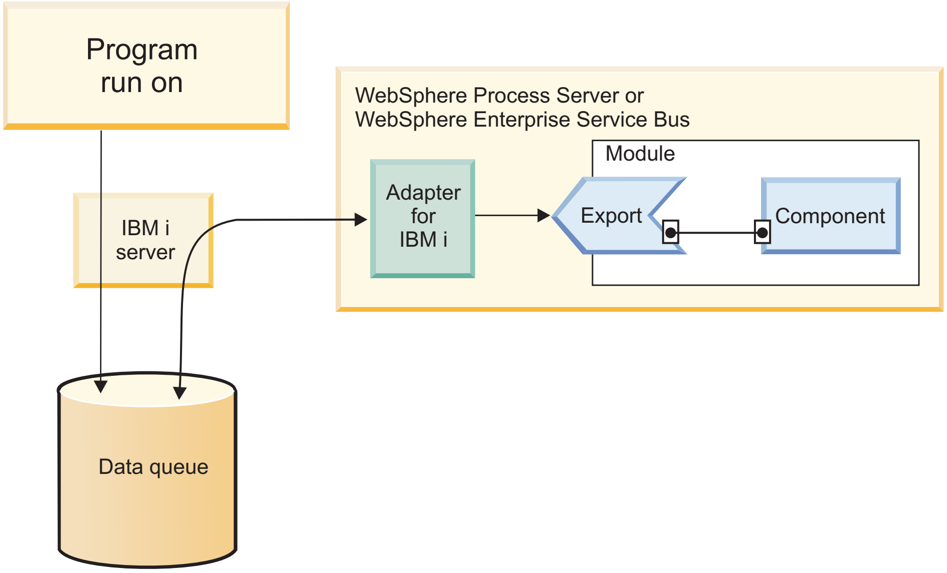


Figure 3. Overview of inbound processing

During each polling interval, the adapter processes a configurable number of messages on the polled queue. For each message, the adapter takes the following actions:

1. Reads a message from the data queue. From a FIFO queue, the adapter fetches the oldest message. From a LIFO queue, adapter fetches the most recent message.
2. Saves the message in the staging queue, to prevent its loss in case of a failure.
3. Creates an event, which records the processing status of the message in the staging queue. If the adapter is configured for assured-once processing, the event is stored in a persistent table called the event store. Otherwise, the events

are stored in memory on the underlying application server where adapter is deployed with the event ID being the key of the message in the staging queue.

4. Processes the event in the event store or memory and delivers it to the client application. To deliver an event, the adapter creates a business object for each message retrieved from the polling queue.
5. Removes the event from the event store and the message from the staging queue after the event is delivered.

To perform inbound processing, you must take the following actions before you deploy the adapter:

- Create the staging queue and fault queue on the IBM i system.
- Copy the control language (CL) program that moves messages from the polled queue to the staging queue. This program is located in the adapter installation directory and must be copied to the IBM i system.
- Specify the location of the staging queue and CL program when you use the external service wizard to configure the adapter.
- Optional: Create an event store to use for assured-once delivery. You can also let the adapter create the event store if one is needed.
- Specify additional event processing options such as event polling, event delivery, event persistence, and logging and tracing.

You can customize event processing when you use the external service wizard to configure the adapter initially or at a later time by using the administrative console of the server to change the activation specification properties. For example, you can control how often the adapter polls for events and how many events it retrieves during each polling period.

The adapter provides sample scripts CL programs I5INBOUND.SAVF and I5ARCHFAIL.SAVF for creating data queues. The CL programs contains the source code (.CLLP files) along with the compiled code (.PGM). These CL programs are located in the WID_installation_dir/ResourceAdapters/IBMi_version/samples/scripts directory, where WID_installation_dir is the installation directory for WebSphere Integration Developer, and version is the version of the adapter, for example, `6.2`.

Assured-once delivery of events

The adapter supports assured-once delivery of events for inbound processing in case of abrupt termination. Assured-once delivery is a way to make sure that events are delivered once, and only once, to the export in the case of a failure. During event processing, the adapter persists the event state in an event store located on the data source. You must set up this data source using WebSphere Process Serverbefore you can create the event store. To use the recovery feature provided by WebSphere Process Server, you set the AssuredOnceDelivery property in the activation specification to `true`. This recovery feature is enabled by default.

The adapter also provides for event persistence using an in-memory representation of the event store. When you use this feature, you do not need to create a JNDI data source or an external event store, and event processing is faster. However, with this feature there is no support for event recovery. In the case of server failure, the in-memory event stores are lost. To prevent the loss of events in the case of server failure, the recommended approach is to use the database event store.

To use the in-memory event persistence capability of the adapter, you must set the AssuredOnceDelivery property to `false`, or the adapter will log a warning message.

Event store

The event store is a persistent cache where event records are saved until they are delivered to the client application.

The adapter uses the event store to keep track of inbound events as they make their way through the system. When a message is removed from the polling queue, the adapter updates the status of the corresponding event in the event store. For recovery purposes, the adapter continually maintains the status of the event in the event store until the event is delivered to the client application. If the adapter is abruptly terminated, the adapter uses the event store to determine which events have and have not been processed. When an event is successfully posted, the entry is deleted from the event store.

Although an event store is not required for the adapter to process inbound requests, the adapter uses one if the assured-once delivery feature is enabled when the adapter is configured using the external service wizard. This allows the adapter to make sure that each event is delivered once, and only once, to the configured client application. If the adapter is not configured to use an event store, similar information is stored in memory on the IBM i system. However, the adapter cannot use the in-memory table to track events in the event that the adapter must be restarted.

Each event store created by the adapter is associated with a specific inbound module. The adapter does not support multiple adapter modules using the same event store.

The table format and content of the event store are described Table 2.

Table 2. Definition of the event store

Column name	Type	Description
EVNTID	Varchar (255)	The unique event ID value for assured-once delivery. The event ID is the key of the message in the staging queue, which is a keyed data queue. If the polled queue is a FIFO or LIFO, the adapter creates a key for messages; the key has the format *AbsolutePathOfQueue+TimeStamp*.
EVNTSTAT	Integer	The event status. The adapter sets this field to the value for a new event when it creates the event. The adapter updates this field as it processes the event. The status can be one of the following values: • 0: Identifies a new event. • -1: Indicates that an error occurred while processing the event. After the event is delivered to the endpoint, the adapter deletes the event from the event table. This column cannot be `null`.
XID	Varchar (255)	The transaction ID for the event. The adapter uses the transaction ID during recovery processing after a delivery failure to ensure that each event is delivered once and only once to the configured client application.

Table 2. Definition of the event store (continued)

Column name	Type	Description
BQTOTAL	Integer	Reserved.
BQPROC	Integer	Reserved.
EVNTDATA	Varchar (255)	Reserved.

Event recovery

You can configure the adapter for inbound processing so that it supports event recovery in case of abrupt termination. When event recovery is specified, the adapter persists the event state in an event recovery table that resides on a data source. Event recovery along with event persistence is not the default; you must specify it by enabling once-only delivery of events during adapter configuration.

Data source

Event recovery for inbound processing requires that a JDBC data source be configured. You use the administrative console to configure the data source. You select a JDBC provider (for example, Derby) and then create a new data source.

Event recovery table

You can create the event recovery table manually, or you can have the adapter create the event table. The value of the EP_CreateTable configuration property determines whether the event recovery table is created automatically. The default value of this property is `True` (create the table automatically).

To create the table manually, use the information provided in the following table.

Table 3. Event recovery table fields

Table field name	Type	Description
EVNTID	VARCHAR(255)	Transaction ID for the tRFC (Transactional Remote Function Call) protocol. The tRFC protocol significantly improves the reliability of the data transfer, but it does not ensure that the order of ALE transactions specified in the application is observed. Event ordering is also affected by the number of event listeners. However, at some point all ALE transactions are transferred.
EVNTSTAT	INTEGER	Event processing status. Possible values are: • 0 (Created) • 1 (Executed) • 3 (In Progress) • -1 (Failed)

Table 3. Event recovery table fields (continued)

Table field name	Type	Description
XID	VARCHAR(255)	An XA resource keeps track of transaction IDs (XIDs) in the event recovery table. The adapter queries and updates that XID field. During recovery, WebSphere Application Server calls the resource adapter, querying it for XA resources, and then does transaction recovery on them. **Note:** The XA resource is used to enable assured once delivery. Make sure the activation specification property Assured Once Delivery is set to `true`.
BQTOTAL	INTEGER	Total number of IDocs in the packet.
BQPROC	INTEGER	Sequence number of the IDoc in the packet that the adapter is currently processing.
EVNTDATA	VARCHAR(255)	Not used.

To use event recovery for multiple endpoints, you must configure a separate event recovery table for each endpoint, although you can use the same data source (for example, Derby) to hold all the event recovery tables.

For retries of the failure of event delivery in conditions such as abrupt termination of the adapter or any process failure like network issue, these failed events will be moved to the failed events queue path by the adapter; as soon as it is restarted.

Business objects

A business object is a structure that consists of data and additional instructions, if any, for processing the data. The adapter uses business objects to represent data queues and RPG programs running on the IBM i system.

How the adapter uses business objects

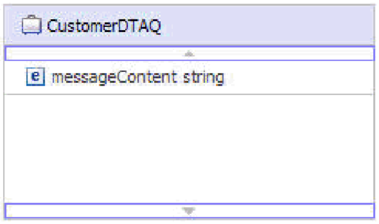
Your component communicates with the adapter using business objects. Module use business objects to access the IBM i data queues or to run RPG programs on the IBM i system. For outbound operations, the adapter converts the business objects in outbound requests into IBM i API calls to access queues and programs. For inbound events, the adapter converts the data in the events into business objects, which are returned to the application.

Note: The operation is not stored in the business object. It is part of the module.

How data is represented in business objects

For data queue business objects

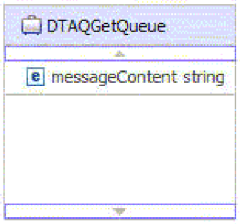
A queue is a data structure for processing work in which the first element added to the queue is the first element processed. This order is referred to as first-in first-out (FIFO). Figure 4 on page 11 shows a business objects to put a message on an IBM i data queue in the business object editor. Each business object has one attribute; for the message content as a string. The absolute path to the data queue (object path) on the IBM i system is not part of the business object. Instead, it is part of the interaction specification properties set in the wizard. Because data queue business objects have no child business objects, all of the attributes are simple attributes.



```
CustomerDTAQ
messageContent string
```

Figure 4. Business object for putting a message from a customer on a data queue

Similarly, Business object for getting a message from a data queue shows a business object used to get a message from an IBM i data queue.



```
DTAQGetQueue
messageContent string
```

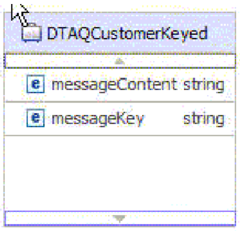
Figure 5. Business object for getting a message from a data queue

Note: If a maximum entry length restriction for the target data queue in the wizard is set, the adapter will generate a business object with the maxLength property appropriately set for the messageContent attribute.

Though the type of data queue (either FIFO or LIFO) may differ, the operation the adapter performs on the business object is transparent. The business object does not indicate what type of data queue the message is being added to or removed from.

For keyed data queue business objects

Figure 6 shows a business object used to remove and process a message from a keyed IBM i data queue in the business object editor. For keyed data queues, the business object has two attributes: one for the message content and one for the message key. The absolute path to the data queue (object path) on the IBM i system is not part of the business object. Instead, it is part of the interaction specification properties set in the wizard. Because data queue business objects have no child business objects, all of the attributes are simple attributes.



```
DTAQCustomerKeyed
messageContent string
messageKey     string
```

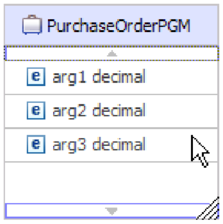
Figure 6. Business object for removing a customer message from a keyed data queue

Note: If a maximum entry length restriction for the target data queue is set in the wizard, the adapter will generate a business object with the maxLength property appropriately set for the messageContent attribute.

For RPG program business objects

Figure 7 shows a business object used to run a RPG program on an IBM i system in the business object editor. To properly invoke a RPG program on an IBM i system, the business object must contain attributes corresponding to each input argument the IBM i system needs to run the program as well as those required for gathering the return values.

For example, a PurchaseOrderPGM business object is used to invoke ″purchaseorder.pgm″ on the IBM i system. The path of the program on the IBM i system is recorded at the interaction specification level.


PurchaseOrderPGM

arg1 decimal

arg2 decimal

arg3 decimal

Figure 7. PurchaseOrderPGM business object

Each PurchaseOrderPGM object contains the arguments required to invoke the program on the IBM i system.

How business objects are created

Business objects can be created through your business object schemas (xsd files) with all the required fields. For example, when manually creating an RPG business object you need to account for application-specific information such as the required fields for invoking an RPG program on the IBM i system. For data queue business objects you would need to include attributes indicating the length of the message content for a data queue business object. For keyed data queue business objects you would need to also include the attribute for the key.

Before xsd files are imported into the wizard, they are called ″pre-defined″ business objects. The wizard connects to the IBM i system, discovers IBM i objects, and displays them to you. You select the objects for which you want to create business objects. After you specify which business objects you want, the wizard then generates business objects to represent the target IBM i objects that you selected. It also generates other artifacts needed by the adapter.

The external service wizard

The external service wizard in WebSphere Adapter for IBM i is used to create services and to generate business objects from the selected objects. The wizard also generates the module and the service artifacts that enable the adapter to run as a Service Component Architecture (SCA) component.

Chapter 2. Planning for adapter implementation

Before using WebSphere Adapter for IBM i, make sure you understand the experience you need and the server environment in which it runs. Learn the considerations for deploying the adapter in your server environment, and find out how to improve the performance and availability of the adapter by using a clustered server environment.

Security

The adapter uses the J2C authentication data entry, or authentication alias, feature of Java 2 security to provide secure user name and password authentication. For more information about security features, see the documentation for WebSphere Process Server or WebSphere Enterprise Service Bus.

Support for protecting sensitive user data in log and trace files

The adapter provides the ability to prevent sensitive or confidential data in log and trace files from being seen by those without authorization.

Log and trace files for the adapter can contain data from your IBM i system, which might contain sensitive or confidential information. Sometimes these files must be seen by individuals without authorization to view sensitive data. For example, a support specialist must use the log and trace files to troubleshoot a problem.

To protect the data in situations like this, the adapter lets you specify whether or not you want to prevent confidential user data from displaying in the adapter log and trace files. You can select this option in the external service wizard or change the HideConfidentialTrace property. When this property is enabled, the adapter replaces the sensitive data with XXX's.

See "Managed connection factory properties" on page 93 for information about this optional property.

The following types of information are considered potentially sensitive data and are disguised:
- The contents of a business object
- The contents of the object key of the event record
- User name and password

The following types of information are not considered user data and are not hidden:
- The contents of the event record that are not part of the event record object key, for example, the XID, event ID, business object name, and event status
- Business object schemas
- Transaction IDs
- Call sequences
- Host name

User authentication

The adapter supports several methods for supplying the user name and password that are needed to connect to the IBM i system. By understanding the features and limitations of each method, you can pick a method that provides the appropriate level of security and convenience for your application.

At run time, the adapter needs to provide the user name and password to connect to the IBM i system. To connect without user intervention, the adapter must access a saved copy of the user information. In a server environment, there are several methods for saving user information. The external service wizard lets you configure the adapter to get the user information by using any of the following methods:

- Adapter properties
- J2C authentication alias

Saving the user name and password in adapter properties is a direct way to provide this information at run time. You provide the user name and password when you use the external service wizard to configure your module. Although directly specifying the user name and password seems the most straightforward method, it has important limitations. Adapter properties are not encrypted; the password is stored as clear text in fields that are accessible to others on the server. Also, when the password changes, you must update the password in all instances of the adapter that accesses that IBM i system, including the adapters embedded in application EAR files as well as adapters that are separately installed on the server.

Using a J2C authentication data entry, or authentication alias, created with the Java Authentication and Authorization Service (JAAS) feature of Java 2 security is a robust, secure way to deploy applications. An administrator creates the authentication alias that is used by one or more applications that need to access a system. The user name and password can be known only to that administrator, who can change the password in a single place when a change is required.

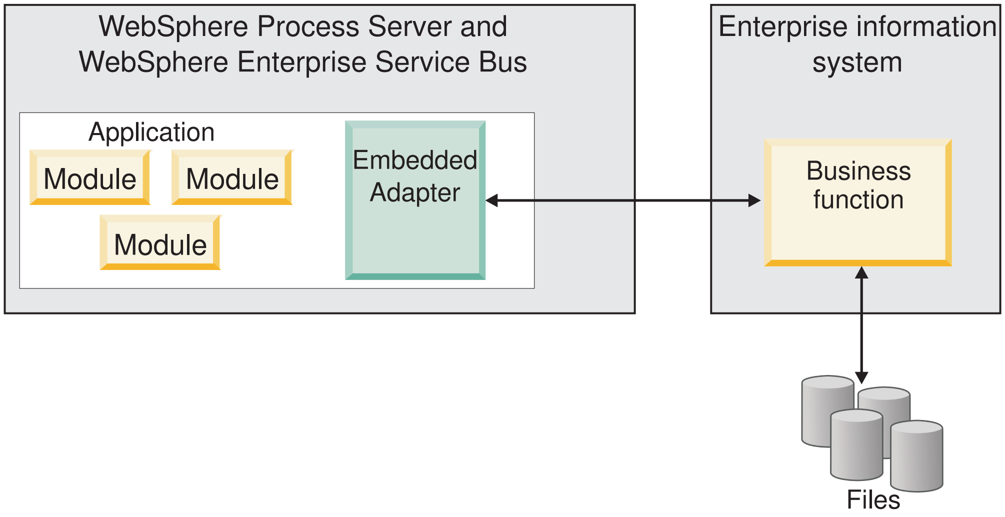
Deployment options

There are two ways to deploy the adapter. You can either embed it as part of the deployed application, or you can deploy it as a stand-alone RAR file. The requirements of your environment will affect the type of deployment option you choose.

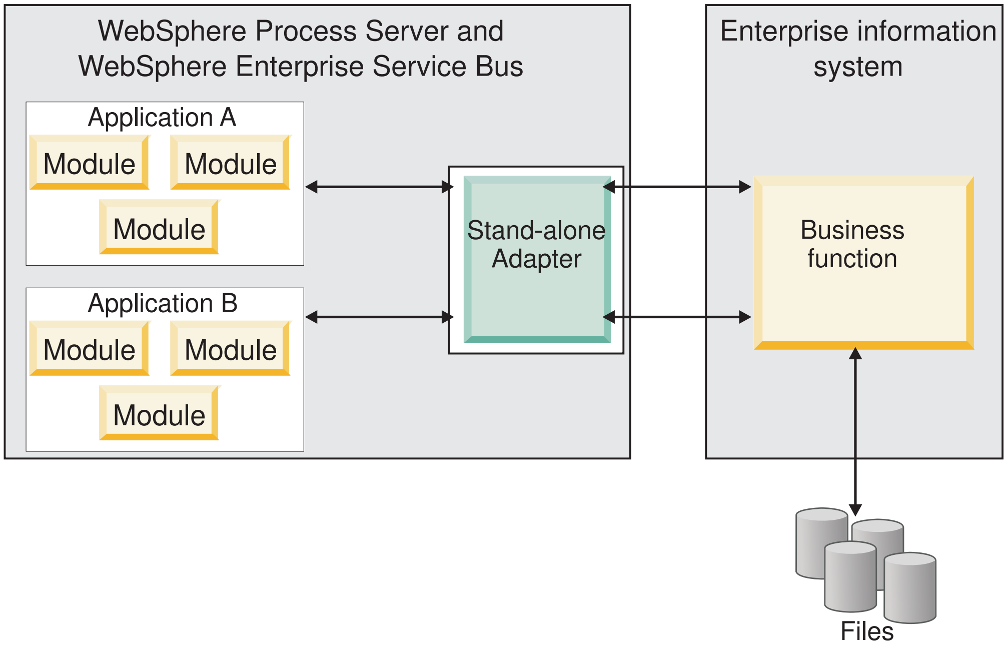
The deployment options are described below:

- **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

An embedded adapter is bundled within an enterprise archive (EAR) file and is available only to the application with which it is packaged and deployed.


WebSphere Process Server and
WebSphere Enterprise Service Bus
Application
Module
Module
Module
Embedded Adapter
Enterprise information system
Business function
Files

A stand-alone adapter is represented by a stand-alone resource adapter archive (RAR) file, and when deployed, it is available to all deployed applications in the server instance.



While creating the project for your application using WebSphere Integration Developer, you can choose how to package the adapter [either bundled with the (EAR) file or as a stand-alone (RAR) file]. Your choice will affect how the adapter is used in the runtime environment, as well as how the properties for the adapter are displayed on the administrative console.

Choosing either to embed an adapter with your application or to deploy the adapter as a stand-alone module depends on how you want to administer the adapter. If you want a single copy of the adapter and do not care about disruption to multiple applications when you upgrade the adapter, then you would be more likely to deploy the adapter as a stand-alone module.

If you plan on running multiple versions, and if you care more about potential disruption when you upgrade the adapter, you would be more likely to embed the adapter with the application. Embedding the adapter with the application allows you to associate an adapter version with an application version and administer it as a single module.

Considerations for embedding an adapter in the application

Take into consideration the following items if you plan on embedding the adapter with your application:

- An embedded adapter has class loader isolation.

  A class loader affects the packaging of applications and the behavior of packaged applications deployed on runtime environments. *Class loader isolation* means the adapter cannot load classes from another application or module. Class loader isolation prevents two similarly named classes in different applications from interfering with each other.
- Each application in which the adapter is embedded must be administered separately.

Considerations for using a stand-alone adapter

Take into consideration the following items if you plan on using a stand-alone adapter:

- Stand-alone adapters have no class loader isolation.

  Because stand-alone adapters have no class loader isolation, only one version of any given Java artifact is run and the version and sequence of that artifact is undetermined. For example, when you use a stand-alone adapter there is only *one* resource adapter version, *one* adapter foundation class (AFC) version, or *one* third-party JAR version. All adapters deployed as stand-alone adapters share a single AFC version, and all instances of a given adapter share the same code version. All adapter instances using a given third-party library must share that library.
- If you update any of these shared artifacts, all applications using the artifacts are affected.

  For instance, if you have an adapter that is working with server version X, and you update the version of the client application to version Y, your original application might stop working.
- Adapter Foundation Classes (AFC) is compatible with previous versions, but the latest AFC version must be in every RAR file that is deployed in a stand-alone manner.

  If more than one copy of any JAR file is in the class path in a stand-alone adapter, the one that is used is random; therefore, they all must be the latest version.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying the module to a clustered server environment. The module is replicated across all servers in a cluster, regardless of whether you deploy the module using a stand-alone or embedded adapter.

WebSphere Process Server or WebSphere Enterprise Service Bus, WebSphere Application Server Network Deployment, and WebSphere Extended 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 Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) container to activate the adapter instance. The JCA container provides a runtime environment for adapter instances. For information about creating clustered environments, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_v61.html.

Using WebSphere Extended Deployment, you can optionally enhance the performance of adapter instances in your clustered environment. WebSphere Extended Deployment extends the WebSphere Application Server Network Deployment capabilities by using a dynamic workload manager instead of a static workload manager, which is used by WebSphere Application Server Network Deployment. The dynamic workload manager can optimize the performance of adapter instances in the cluster by dynamically balancing the load of the requests. This means that application server instances can be automatically stopped and started based on the load variations, allowing machines with different capacities and configurations to evenly handle load variations. For information on the benefits of WebSphere Extended Deployment, see the following link: http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1/index.jsp.

High availability for inbound processes

Inbound processes are based on events triggered as a result of updates to data in the IBM i system. WebSphere Adapter for IBM i is configured to detect updates by polling an event table. The adapter then publishes the event to its endpoint.

When you deploy a module to a cluster, the JCA container checks the enableHASupport resource adapter property. If the value for the enableHASupport property is true, which is the default setting, all of the adapter instances are registered with the HAManager with a policy 1 of N. This policy means that only one of the adapter instances starts polling for events. 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.

Important: Do not change the setting of the enableHASupport property.

High availability for outbound processes

In clustered environments, multiple adapter instances are available to perform outbound process requests. Accordingly, if your environment has multiple applications that interact with WebSphere Adapter for IBM i for outbound requests, then you might improve performance by deploying the module to a clustered environment. In a clustered environment, multiple outbound requests can be processed simultaneously, as long as they are not attempting to process the same record.

If multiple outbound requests are attempting to process the same record, such as a Customer address, the workload management capability in WebSphere Application Server Network Deployment distributes the requests among the available adapter instances in the sequence they were received. As a result, these types of outbound requests in a clustered environment are processed in the same manner as those in a single server environment: one adapter instance processes only one outbound request at a time. For more information on workload management, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Chapter 3. Samples and tutorials

To help you use WebSphere Adapters, samples and tutorials are available from the Business Process Management Samples and Tutorials Web site.

You can access the samples and tutorials in either of the following ways:

- From the welcome page that opens when you start WebSphere Integration Developer. To see samples and tutorials for WebSphere Adapter for IBM i, click **Retrieve**. Then browse the displayed categories to make your selections.
- At this location on the Web: Business Process Management Samples and Tutorials http://publib.boulder.ibm.com/bpcsamp/index.html.

Chapter 4. Configuring the module for deployment

To configure the adapter so that it can be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, use WebSphere Integration Developer to create a module, which is exported as an EAR file when you deploy the adapter.

Road map for configuring the module

Before you can use WebSphere Adapter for IBM i in a runtime environment, you must configure the module. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

You configure the module for the adapter to use by using WebSphere Integration Developer. The following figure illustrates the flow of the configuration task, and the steps that follow the figure describe this task at a high level only. See the topics following this road map for the details on how to perform each of these steps.

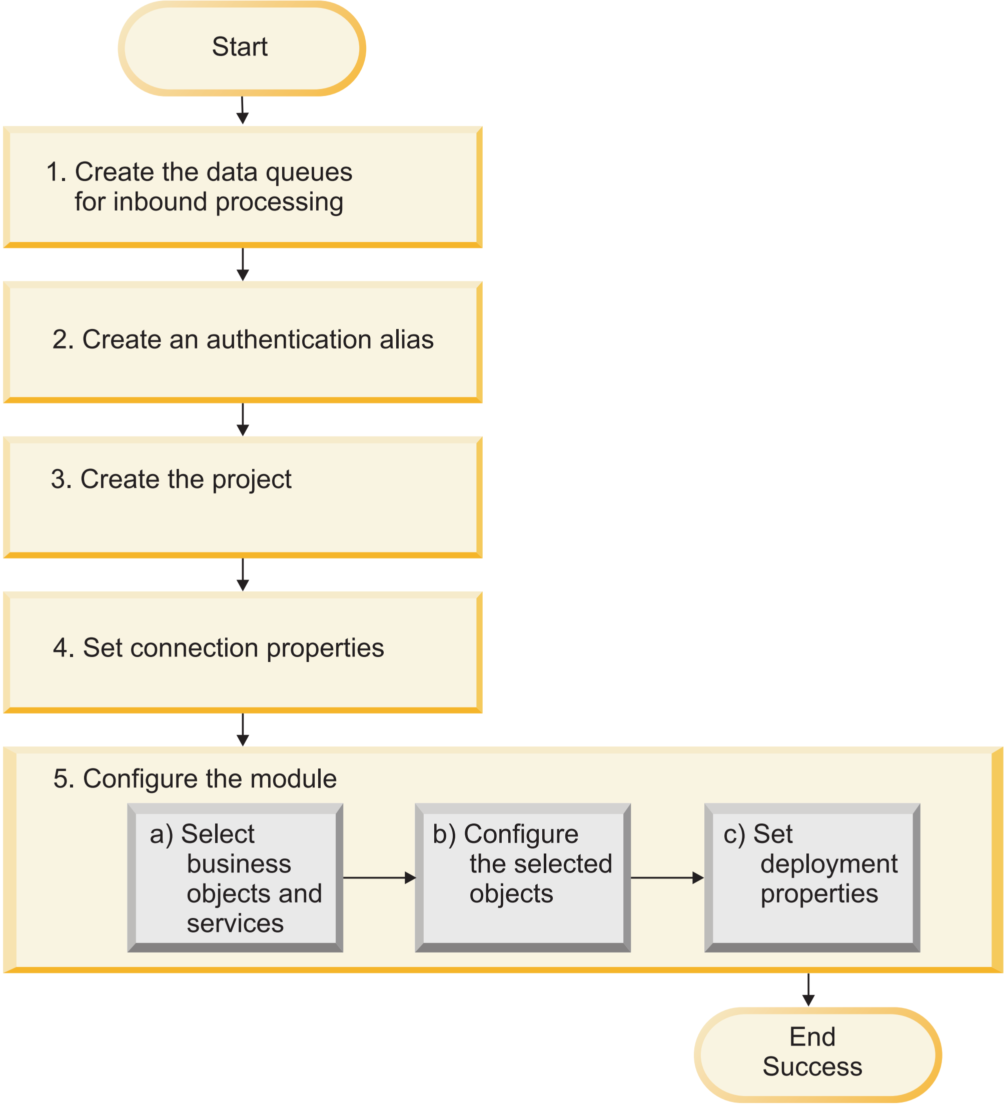

Start
1. Create the data queues for inbound processing
2. Create an authentication alias
3. Create the project
4. Set connection properties
5. Configure the module
a) Select business objects and services
b) Configure the selected objects
c) Set deployment properties
End Success

Figure 8. Road map for configuring the module

Configuring the module for deployment

This task consists of the following step:

1. Create data queues for inbound processing.
2. Create an authentication alias to access the IBM i system with an encrypted password. This step is optional, depending on your policy for handling passwords and IDs. You perform this step using the server.
3. Create the project. First, start the external service wizard in WebSphere Integration Developer to begin the process of creating and deploying a module. The wizard creates a project that is used to organize the files associated with the module.
4. Set connection properties that the external service wizard needs to connect to the IBM i system for discovery of objects and services.
5. Configure the module for inbound or outbound processing by using the external service wizard to find and select business objects and services from the IBM i system, and to generate business object definitions and related artifacts.

a. Select business objects and services for inbound or outbound processing from the business integration components discovered by the external service wizard.
b. Configure the selected objects by specifying operations and other properties that apply to all of the business objects.
c. Set deployment properties that the adapter uses to connect to the IBM i system at run time. Then, generate the service by using the external service wizard to save the new module, which contains the business object or objects you configured, the import or export file, and the service interface.

Creating the data queues for inbound processing

To configure the server for inbound processing, you must create data queues, which are used to complete a specific service. To help you do this, some sample scripts are provided as control language (CL) programs and located in the installation directory for WebSphere Integration Developer.

About this task

Perform this task only if you need to configure the IBM i system for inbound processing of data queues.

Procedure

1. Create the data queues. To help you do this, sample scripts for creating the data queues are provided: I5INBOUND.SAVF and I5ARCHFAIL.SAVF. The CL programs (I5INBOUND.SAVF and I5ARCHFAIL.SAVF) contains the source code (.CLLP files) along with the compiled code (.PGM). These CL programs are located in the *WID_installation_dir*/ResourceAdapters/IBMi_*version*/ samples/scripts directory, where *WID_installation_dir* is the installation directory for WebSphere Integration Developer, and *version* is the version of the adapter, for example, `6.2`. The files I5INBOUND.SAVF and I5ARCHFAIL.SAVF are in compressed format (.SAVF) of IBM i system. The file I5INBOUND.SAVF contains the script and the compiled code for polling messages. The file I5ARCHFAIL.SAVF contains the script and compiled code for archiving failed messages to failed events queue.
2. Extract the CL programs for inbound from the sample build:
   a. On the target IBM i system, enter the command CRTSAVF to create a save (SAVF) file, and name it I5INBOUND, for example: CRTSAVF FILE(TEST/I5INBOUND).
   b. Copy or use FTP to transfer the I5INBOUND.SAVF file to the target IBM i system in binary mode.
   c. Extract the SAVF by entering the command RSTOBJ, for example, RSTOBJ OBJ(I5INBOUND) SAVLIB(RAJA1) DEV(*SAVF) SAVF(TEST/I5INBOUND) RSTLIB(TEST). Also, note that in this example, the adapter build is created from RAJA1.lib.
   d. Optional: Delete the SAVF file after you have extracted it by entering the command DLTF.

Results

The IBM i system is configured and the data queues is available for inbound processing.

Creating an authentication alias

An authentication alias is a feature that encrypts the password used by the adapter to access the IBM i system. The adapter can use it to connect to the IBM i system instead of using a user ID and password stored in an adapter property.

Before you begin

To create an authentication alias, you must have access to the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus. You must also know the user name and password to use to connect to the IBM i system.

The following procedure describes how to access administrative console through WebSphere Integration Developer. If you are using the administrative console directly (without going through WebSphere Integration Developer), log in to the administrative console and skip to step 2.

About this task

Using an authentication alias eliminates the need to store the password in clear text in an adapter configuration property, where it might be visible to others.

To create an authentication alias, use the following procedure.

Procedure

1. Start the administrative console:

   To start the administrative console through WebSphere Integration Developer, perform the following steps:

   a. In the Business Integration perspective of WebSphere Integration Developer, click the **Servers** tab.

   b. If the server does not show a status of **Started**, right-click the name of the server (for example, **WebSphere Process Server**) and click **Start**. Wait until the status of the server is `Started`.

   c. Right-click the name of the server and click **Run administrative console**.

   d. Log on to the administrative console by clicking **Log in**. If your administrative console requires a user ID and password, type the ID and password first and click **Log in**.
2. In the administrative console, click **Security** → **Secure administration, applications, and infrastructure**.
3. Under **Authentication**, click **Java Authentication and Authorization Service** → **J2C Authentication data**.
4. Create an authentication alias

   a. From the list of J2C authentication aliases that is displayed, click **New**.

   b. In the **Configuration** tab, type the name of the authentication alias in the **Alias** field.

   c. Type the user ID and password that are required to establish a connection to the IBM i system.

   d. Optional: Type a description of the alias.

   e. Click **OK**.

      The newly created alias is displayed.

The full name of the alias contains the node name and the authentication alias name you specified. For example, if you create an alias on the node widNode with the name ProductionServerAlias, then the full name is widNode/ProductionServerAlias. This full name is the one you use in subsequent configuration windows.

f. Click **Save**.

Results

You have created an authentication alias, which you specify when you configure the adapter properties later in the wizard.

Creating the project

To begin the process of creating and deploying a module, you start the external service wizard in WebSphere Integration Developer. The wizard creates a connector project, which is used to organize the files associated with the module.

Before you begin

Make sure you have gathered the information you need to establish a connection to the IBM i system. For example, you need the name or IP address of the IBM i system and the user ID and password needed to access it.

About this task

If you have an existing project, you can use it instead of creating a new one. Select it before you start the wizard.

Procedure

1. To start the external service wizard, go to the Business Integration perspective of WebSphere Integration Developer, and then click **File** → **New** → **External Service**.
2. In the External Service window, select **Adapters** and click **Next**.
3. In the Select an Adapter window, select **IBM WebSphere Adapter for IBM i (IBM : *version*)**, where *version* is the version of the adapter you want to use, for example, 6.2.
4. Click **Next**.
5. In the Adapter Import window, accept the default project name in **Connector project** or type a different name.
6. In **Target runtime**, select the type of server where you will deploy the module. The wizard creates the artifacts that are appropriate to that server.
7. Click **Next**. The Required Files and Libraries window is displayed.

Results

A new connector project is created, which contains the adapter RAR file. The project is listed in the Business Integration perspective.

What to do next

Continue working in the external service wizard. The next step is to add database-specific files to the project.

Setting connection properties for the external service wizard

For the external service wizard to be able to discover business objects on the IBM i system, you must specify the connection properties.

About this task

The external service wizard requires these properties to connect to the IBM i system for discovery and for creating the service description. For more information about the properties, see "Connection properties for the wizard" on page 89.

Procedure

1. Start the external service wizard and in the Processing Direction window, select **Outbound** or **Inbound**, and then click **Next**.
2. In the Discovery Configuration window, specify the connection properties for the wizard to use to connect to the IBM i system.
   a. In the **Host name** field, specify the host name or IP address of the IBM i system.
   b. In the **User name** field, enter the user name used to connect to the server from the wizard.
   c. In the **Password** field, enter the password used to connect to the server from the wizard
   d. In the **Path to library on IBM i for object discovery** field, enter the path to the library on the IBM i system, for example, /QSYS.LIB/
   e. In the **Object type to discover** list, select an object type. You have an option to either select data queue or Report Program Generation from the object type to discover list.
   f. In the **Prefix for business object names** list, type a string to be placed at the beginning of business object names. For example, if you select a business object called FIFOQ and you type `Test` for **Prefix for business object names**, the business object created by the wizard will be called TestFIFOQ.

   Connection information from this window is used by the adapter to connect to the IBM i system and discover business objects. Later in the wizard, you can specify different connection information or an authentication alias to use at run time.
3. Optional: To enable bidirectional support for the adapter at run time:
   a. Click **Advanced**.
   b. In the **Bidi properties** list, select **Bidi transformation**.
   c. Set the ordering schema, text direction, symmetric swapping, character shaping, and numeric shaping properties to control how bidirectional transformation is performed.
4. Optional: To change the location of the log files for the wizard or the amount of information included in the logs, click **Change logging properties for wizard**, and then provide the following information:
   - In **Log file output location**, specify the location of the log file for the wizard.
   - In **Logging level**, specify the severity of errors that you want logged.

   This log information is for the wizard only; at run time, the adapter writes messages and trace information into the standard log and trace files for the server.
5. Click **Next**.

Results

The external service wizard connects to the database and displays the Object Discovery and Selection window.

What to do next

Continue working in the wizard. Locate the objects on the IBM i system for which you want the wizard to create business objects.

Configuring the module for outbound processing

To configure a module to use the adapter for outbound processing, use the external service wizard in WebSphere Integration Developer to find and select business objects and services from the IBM i system, and to generate business object definitions and related artifacts.

Selecting and configuring business objects

Use the list of objects discovered by the external service wizard to select the IBM i server business objects that you need to access in your module. Then provide configuration information for your new business objects.

Before you begin

Before you can set the properties in this section, you must have created your adapter module. It should be displayed in WebSphere Integration Developer below the adapter project. For more information on creating the adapter project, see "Creating the project" on page 25

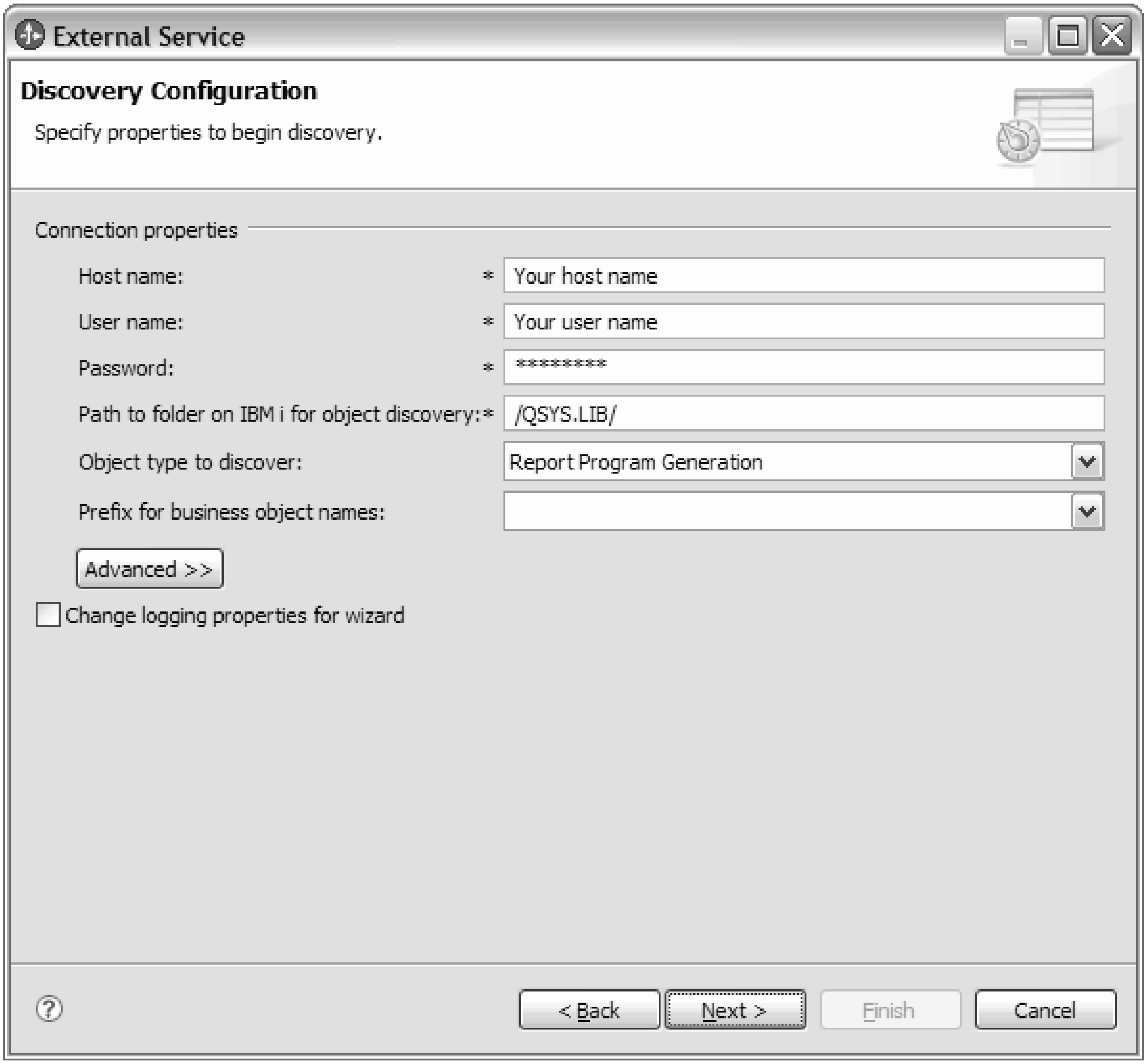
About this task

The Object Discovery and Selection window allows you to select and configure objects in any order. You can mix objects from the various nodes of the **Discovered objects** list such as Program Call Markup Language (PCML).

To set connection properties and to search for and select business objects, follow this procedure. For more information on any of the properties in this topic, see the reference topic devoted to manage connection factory properties in this documentation.

Procedure

1. In the Processing Direction window, select **Outbound** (to send data to the IBM i system from the adapter), and click **Next**.
2. In the Discovery Configuration window, type the host name, user name, password and the path to folder on IBM i for object discovery. Select either data queue or Report Program Generation as per your requirements from the **Object type to discover** list. For example, the following figure shows the operations available for the report program generation business objects. Click **Next**.



Discovery Configuration

Specify properties to begin discovery.

Connection properties

Host name: * Your host name

User name: * Your user name

Password: * *********

Path to folder on IBM i for object discovery:* /QSYS.LIB/

Object type to discover: Report Program Generation

Prefix for business object names:

Advanced >>

□ Change logging properties for wizard

Figure 9. Discovery Configuration window

3. In the Object Discovery and Selection window, select an object from the **Discovered objects** pane.

   Note: If you mouse over a report program generation business object, the wizard will display the report program generation name and the PCML type.
4. Click the > (Add) button. The selected objects are displayed in the Selected objects pane.

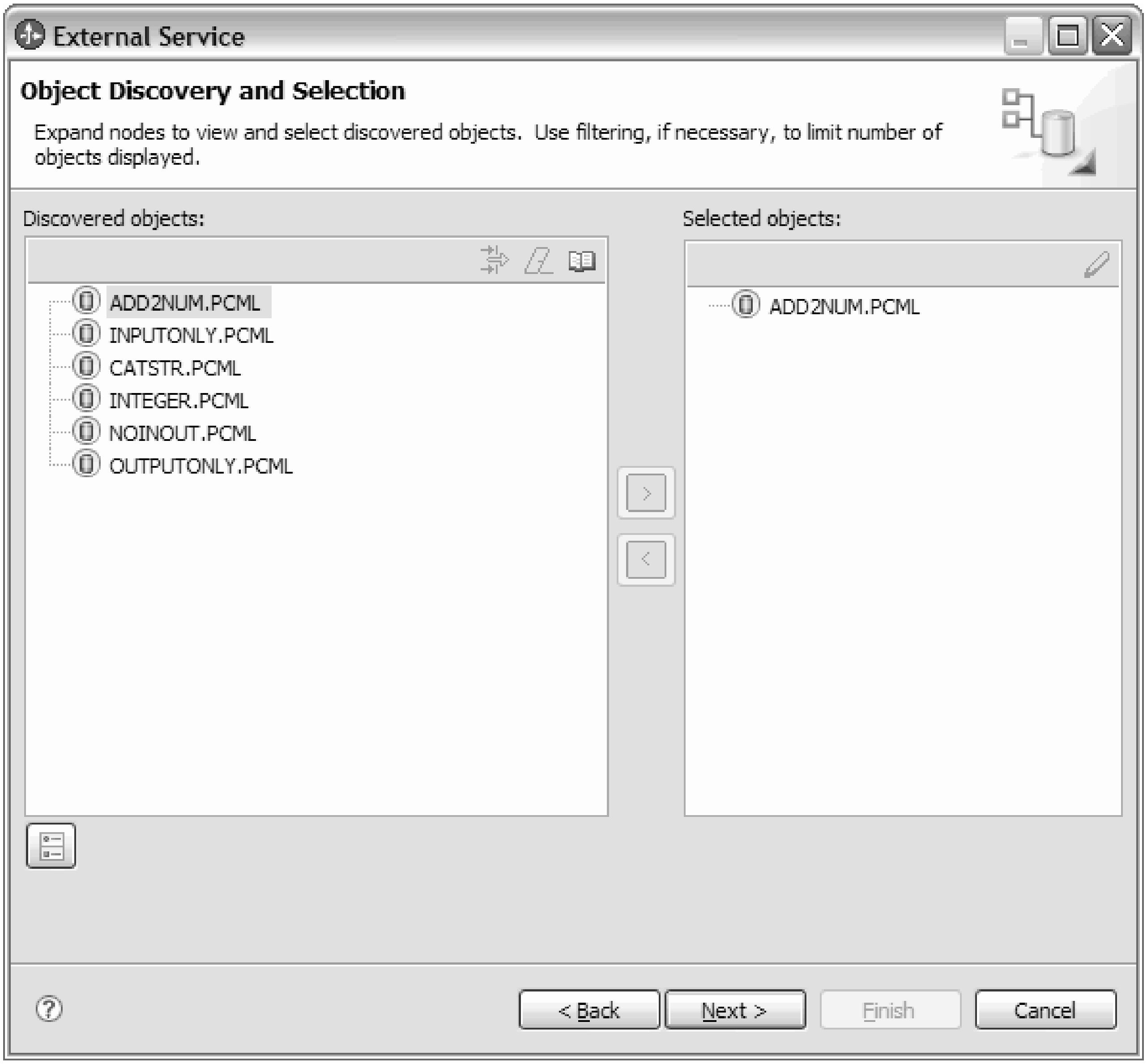


Figure 10. Object Discovery and Selection window

5. After you have selected the business objects you want to add to the module, click **Next**.

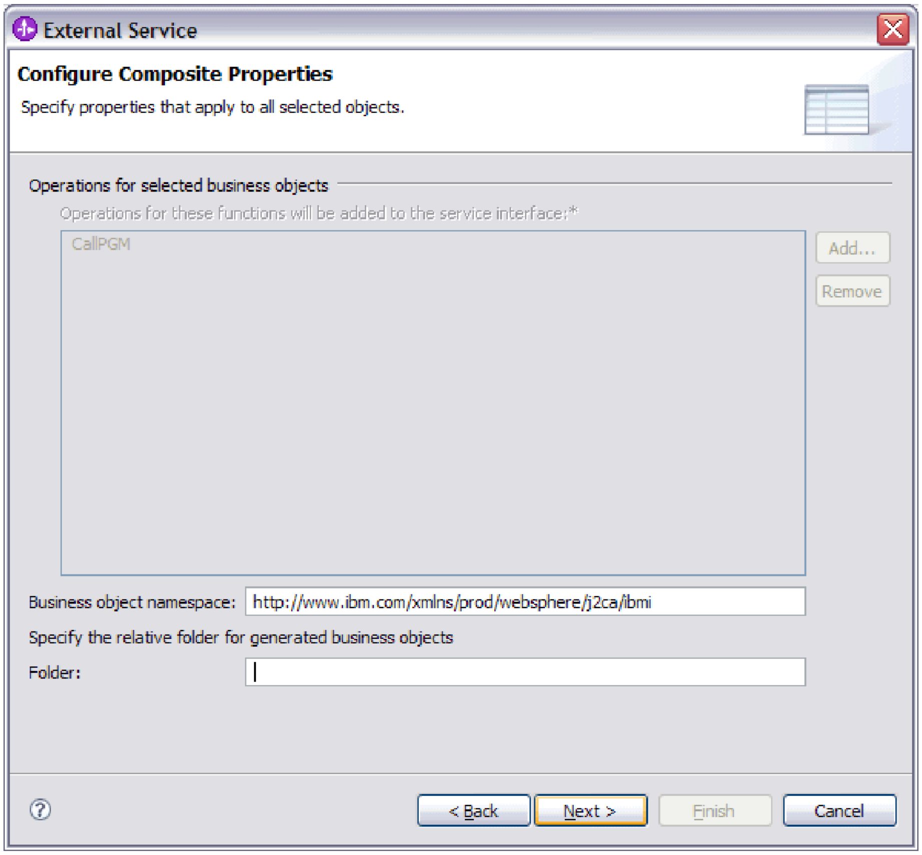
As long as you do not exit from the wizard, you can work iteratively to select and configure the business objects you need in your module. After you exit the wizard, you cannot use it to change or add objects to an existing module, so be careful to understand the requirements of the program that uses the business objects before you start the wizard.

Configuring the selected objects

After you have selected metadata object in the external service wizard, you need to specify properties that apply to them.

Procedure

1. When the **Selected objects** list in the Object Discovery and Selection window contains all of the business objects you want to use in your application, click **Next**.
2. The Configure Composite Properties window displays the operation name for selected business objects. For data queue the operation names are `PutQueue` and `GetQueue`. For report program generation, the operation name is `CallPGM`.



External Service

Configure Composite Properties

Specify properties that apply to all selected objects.

Operations for selected business objects

Operations for these functions will be added to the service interface:*

CallPGM

Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi

Specify the relative folder for generated business objects

Folder:

Figure 11. Configure composite properties window

3. In the **Business object namespace** field, either accept the default namespace or type the full name of another namespace.

   The namespace is prepended to the business object name to keep the business object schemas logically separated. For more information about this property, see the reference topic devoted to managed connection factory properties in this documentation.
4. Optional: In the **Folder**, type the relative path to the folder where the generated business objects are to be stored. Click **Next**.

Results

You have provided information that applies to all business objects in the module.

What to do next

Continue working in the wizard. The next step is to specify deployment information to use at run time and information for saving the service as a module.

Setting deployment properties and generating the service

After you select and configure business objects for your module, use the external service wizard to configure properties that the adapter uses to connect to an IBM i system. The wizard creates a new business integration module where all the artifacts and property values are saved.

Before you begin

This task is performed using the Service Generation and Deployment and Service Location Properties windows of the external service wizard.

The connection properties in this task are initialized to the values that the wizard uses to connect to an IBM i system. To configure the module to use other values than those used to connect to the IBM i system, change values such as Host name, User name and Password here.

Important: If the module contains an interface or business object with the same name as any you are now configuring, the original interface or business object in the module is replaced by the new version.

About this task

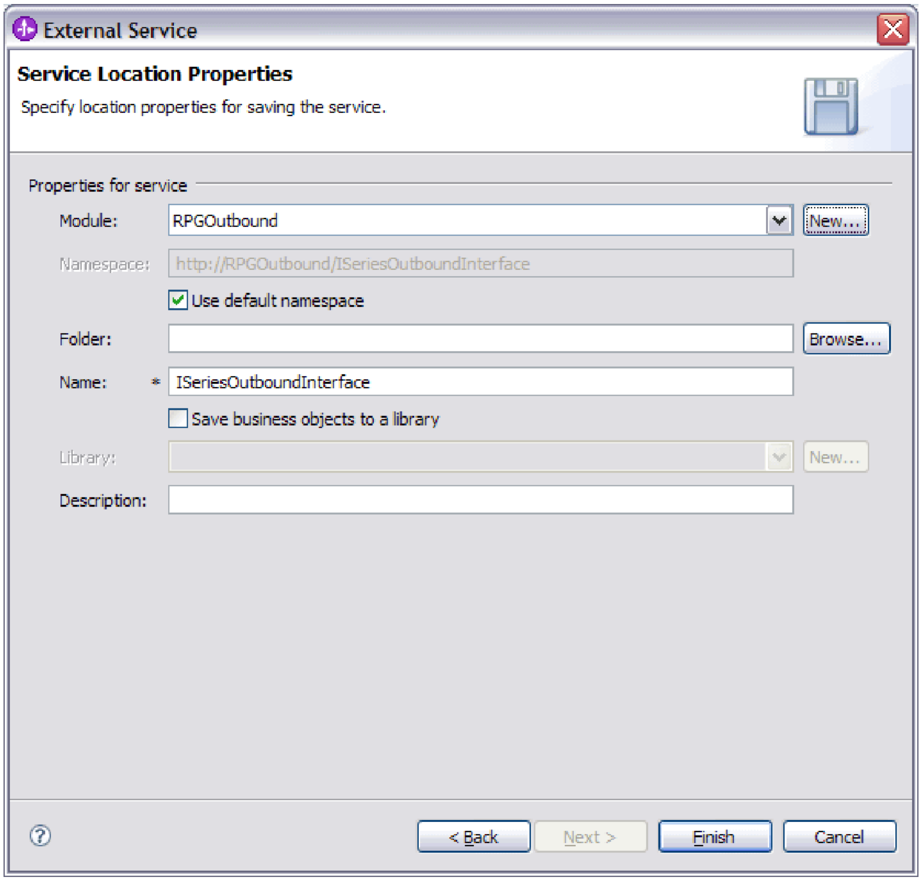
To generate artifacts for your module follow this procedure:

Procedure

1. In the Service Generation and Deployment Configuration window, click **Edit Operations** to review the names of or add a description for the operations for the business objects you are creating.
2. Specify how you want the adapter to connect to IBM i system.
3. To use a J2C authentication alias, specify whether you want the adapter to use a **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** and type the name of the alias in **J2C Authentication Data Entry** field. You can either specify an existing authentication alias or create one at any time before deploying the module. The name is case-sensitive and includes the node name.
4. Optional: To use the user name and password specified in an existing Java Naming and Directory Interface (JNDI) data source on the server:
   a. Clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
   b. Click **Advanced**.
   c. Expand **Advanced connection configuration**.
   d. In **Datasource JNDI name** field, type the name of an existing JNDI data source.
5. In **Deploy connector project**, specify whether to include the adapter files in the module. Choose one of the following values:
   - **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

6. If you selected **On server for use by multiple adapters** in the previous step, specify how you want to set the connection properties at runtime.
   - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same IBM i system using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source:
     a. In **Connection properties**, select **Use predefined connection properties**.
     b. In **JNDI Lookup Name**, type the name of the JNDI data source for an existing managed connection factory or activation specification.
     c. Click **Next**.
   - If this is the first application that connects to the database with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.
7. Optional: Review and if necessary, change the values of the required connection properties. The fields are initialized with the connection information you specified when you started the wizard. You can change the values to specify a different user name and password at run time. You can also connect to an alternate IBM i system, although the schema names must be the same in both databases. The format of the connection properties are database-specific.
8. Optional: Specify advanced properties by clicking **Advanced**. If the module contains an interface or business object with the same name as any you are now configuring, the original interface or business object in the module is replaced by the new version
   - **Advanced properties**
     a. In the **User name** field, type the user name used to connect to the server from the wizard.
     b. In the **Password** field, type the password used to connect to the server from the wizard
   - **Logging and tracing**
     – If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance. For more information about the Adapter ID property, see "Adapter ID (AdapterID)" on page 93.
9. Click **Next**. The Service Location Properties window is displayed.



Service Location Properties

Specify location properties for saving the service.

Properties for service

Module: RPGOutbound

Namespace: http://RPGOutbound/ISeriesOutboundInterface

☑ Use default namespace

Folder:

Name: * ISeriesOutboundInterface

☐ Save business objects to a library

Library:

Description:

10. In the Service Location Properties window, either select an existing module or create a new one.
    - To use an existing module, select its name from the **Module** list.

      Important: If the module contains an interface or business object with the same name as any you are now configuring, the original interface or business object in the module is replaced by the new version.
    - To create a new module:
      a. Click **New**.
      b. In the Integration Project window, select **Create a module project** and click **Next**.
      c. In the Module window, type a name for the module. For example, ISeriesOutboundInterface.
      d. If you want the service description files (the .import and .wsdl files) to be located in the default folder in the module, leave **Use default location** selected. If you want to specify a different folder in the module, clear the option and then click **Browse** to specify a different folder in **Location**.
      e. If you want the module to be automatically opened in the assembly diagram in WebSphere Integration Developer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
      f. Click **Finish** to create the new module.
11. Specify the namespace you want to use for your business objects.
    - If you want the business objects in the module to use the default derived namespace, leave **Use default namespace** check box selected.

- To specify a different namespace, clear the option and type a different value in **Namespace**.

12. Specify the folder within the new module where the service description will be saved. In **Folder**, type the folder name or browse to an existing folder. If you do not specify a folder name, the artifacts (the import, XSD, and WSDL files) are stored in the root folder of the module (the folder with the module name).
13. In **Name**, accept the default import name or type a different name.
14. If you want to save business objects in a library where they can be used by other modules, select **Save business objects to a library** and specify the location of the library in **Library**.
15. In **Description**, type a descriptive comment about the module.
16. When you are finished setting properties, click **Finish**.
17. If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The module is created in the project and artifacts are generated.

Configuring the module for inbound processing

To configure a module to use the adapter for inbound processing, use the external service wizard in WebSphere Integration Developer to find and select business objects and services from the IBM i system, and to generate business object definitions and related artifacts.

Selecting and configuring business objects

Use the list of objects discovered by the external service wizard to select the IBM i server business objects that you need to access in your module. Then provide configuration information for your new business objects.

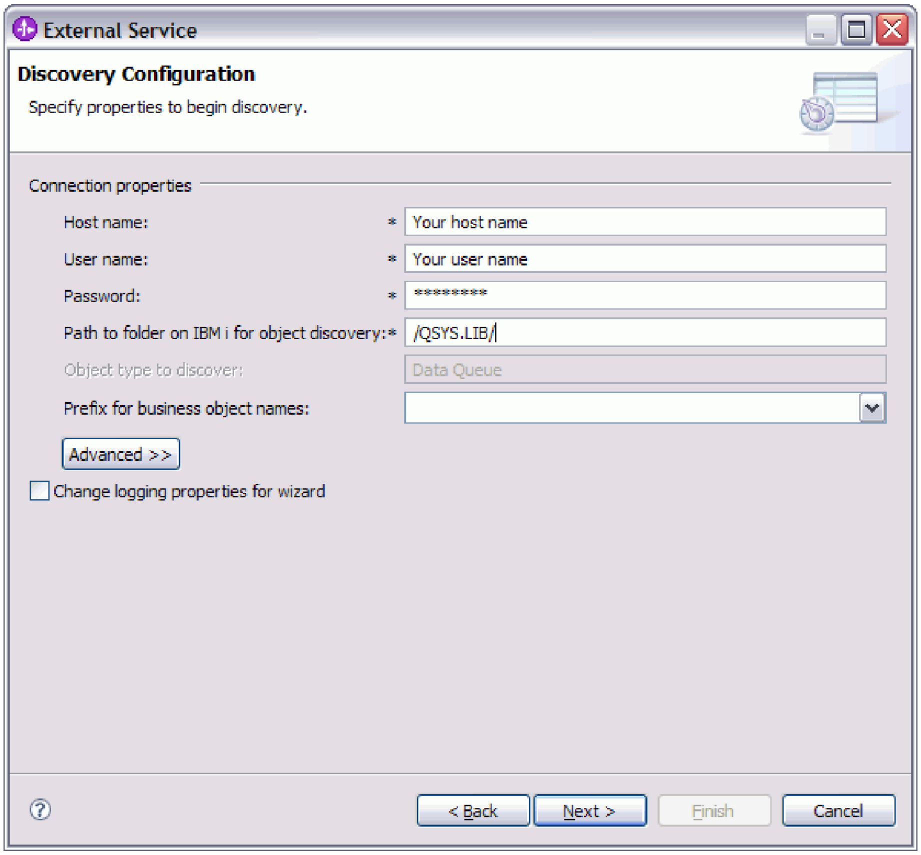
Before you begin

Before you can set the properties in this section, you must have created your adapter module. It should be displayed in WebSphere Integration Developer below the adapter project. For more information on creating the adapter project, see "Creating the project" on page 25

About this task

To set connection properties, follow this procedure. For more information on any of the properties in this topic, see the reference topic devoted to activation specification properties in this documentation.

1. In the Processing Direction window, select **Inbound** (to send data to the adapter from the IBM i system), and click **Next**.
2. Optional: For last-in first-out (LIFO) or first-in first-out (FIFO) data queue polling, in the Discovery Configuration window, as shown in the following figure, type the values in the host name, user name, password and the path to folder on IBM i for object discovery. Click **Next**.


External Service

Discovery Configuration

Specify properties to begin discovery.

Connection properties

Host name: * Your host name

User name: * Your user name

Password: * *********

Path to folder on IBM i for object discovery:* /QSYS.LIB/

Object type to discover: Data Queue

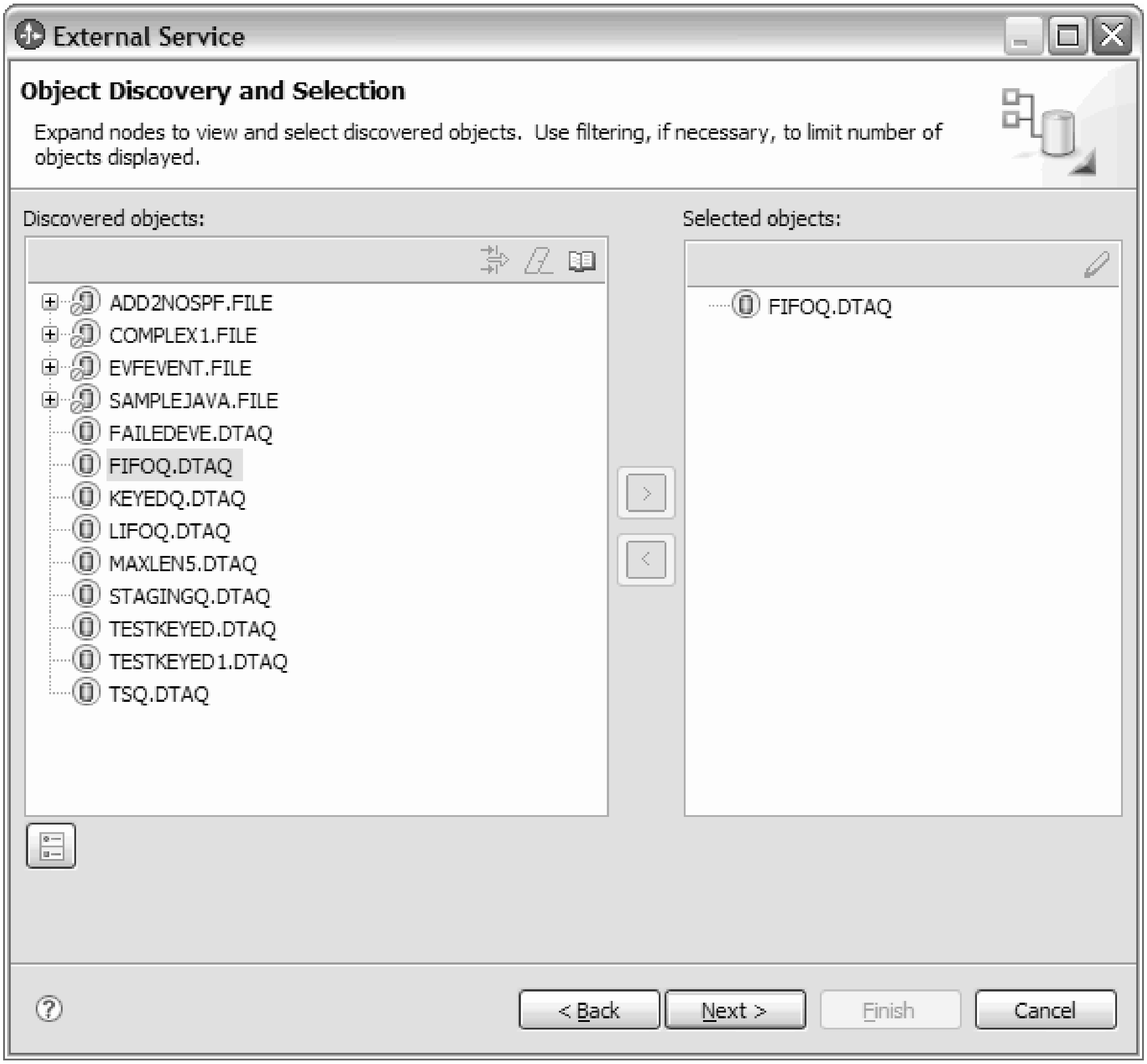
Prefix for business object names:

Figure 12. Discovery Configuration window

Note: The **Object type to discover** field is not available for editing for a simple LIFO or FIFO data queue polling.

3. Optional: In the Object Discovery and Selection window, select an object from the **Discovered objects** pane. For LIFO or FIFO data queue polling, select `LIFOQ.DTAQ` or `FIFOQ.DTAQ`. For example, as shown in the following figure, select `FIFOQ.DTAQ`. Click **Next**.
4. Click the > (Add) button. The added objects are displayed in the Selected objects pane. Click **Next**.

   Note: Only one data queue can be polled in an inbound operation and `FIFOQ.DTAQ` is already added to the selected objects.



External Service

Object Discovery and Selection

Expand nodes to view and select discovered objects. Use filtering, if necessary, to limit number of objects displayed.

Figure 13. Object Discovery and Selection window

5. Optional: For a keyed data queue polling, in the Object Discovery and Selection window, select an object from the **Discovered objects** list. Click **Next**.
6. Select `KEYEDQ.DTAQ` from the Discovered objects pane. Click the > (Add) button. The added objects are displayed in the Selected objects pane. Click **Next**.

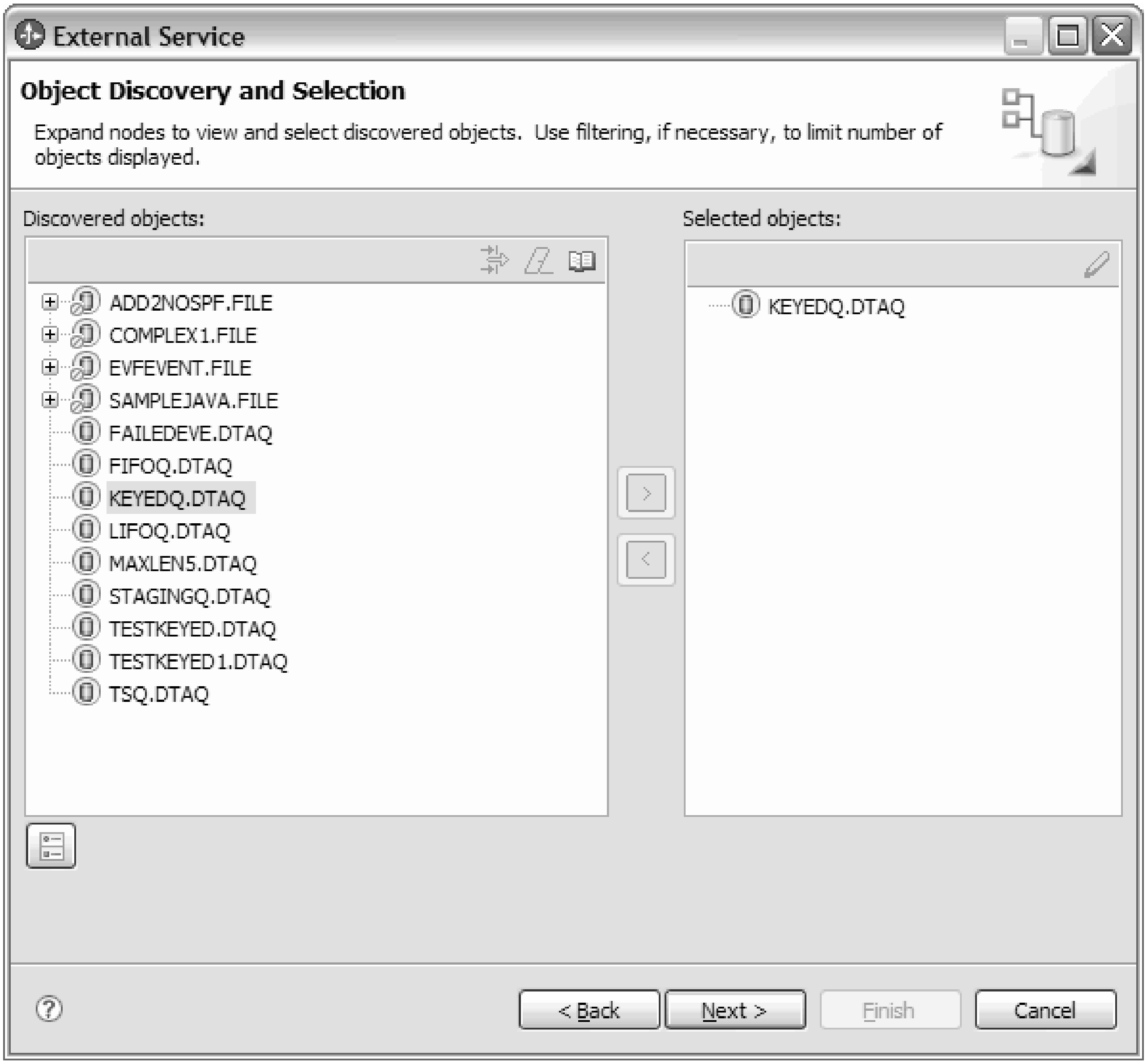


Figure 14. Object Discovery and Selection window

As long as you do not exit from the wizard, you can work iteratively to select and configure the business objects you need in your module. After you exit the wizard, you cannot use it to change or add objects to an existing module, so be careful to understand the requirements of the program that uses the business objects before you start the wizard.

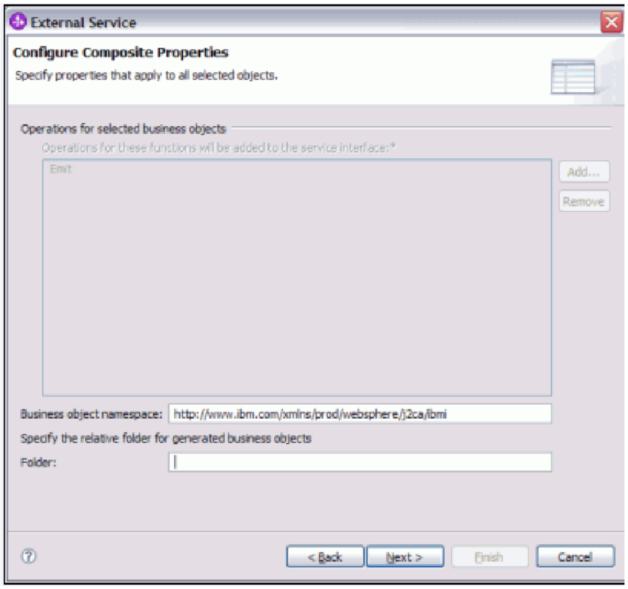
Setting global properties for operations

After you have selected metadata object in the external service wizard, you need to specify properties that apply to all business objects.

Procedure

1. When the **Selected objects** list in the Object Discovery and Selection window contains all of the business objects you want to use in your application, click **Next**.

2. The Configure Composite Properties window displays the operation name, which is `Emit`. The emit operation is the only operation available during inbound processing.



External Service

Configure Composite Properties

Specify properties that apply to all selected objects.

Operations for selected business objects

Operations for these functions will be added to the service interface:*

Emit

Business object namespace: http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi

Specify the relative folder for generated business objects

Folder:

3. Optional: In the **Business object namespace** field, either accept the default namespace or type the full name of another namespace.

   The namespace is prepended to the business object name to keep the business object schemas logically separated. For more information about this property, see the reference topic devoted to activation specification properties in this documentation.
4. Optional: In the **Folder** field, type the relative path to the folder where the generated business objects are to be stored.
5. Click **Next**.

Results

You have provided information that applies to all business objects in the module.

What to do next

Continue working in the wizard. The next step is to specify deployment information to use at run time and information for saving the service as a module.

Setting deployment properties and generating the service

After you select and configure business objects for your module, use the external service wizard to configure properties that the adapter uses to connect to an IBM i system. The wizard creates a new business integration module where all the artifacts and property values are saved.

Before you begin

This task is performed using the Service Generation and Deployment Configuration and Service Location Properties windows of the external service wizard.

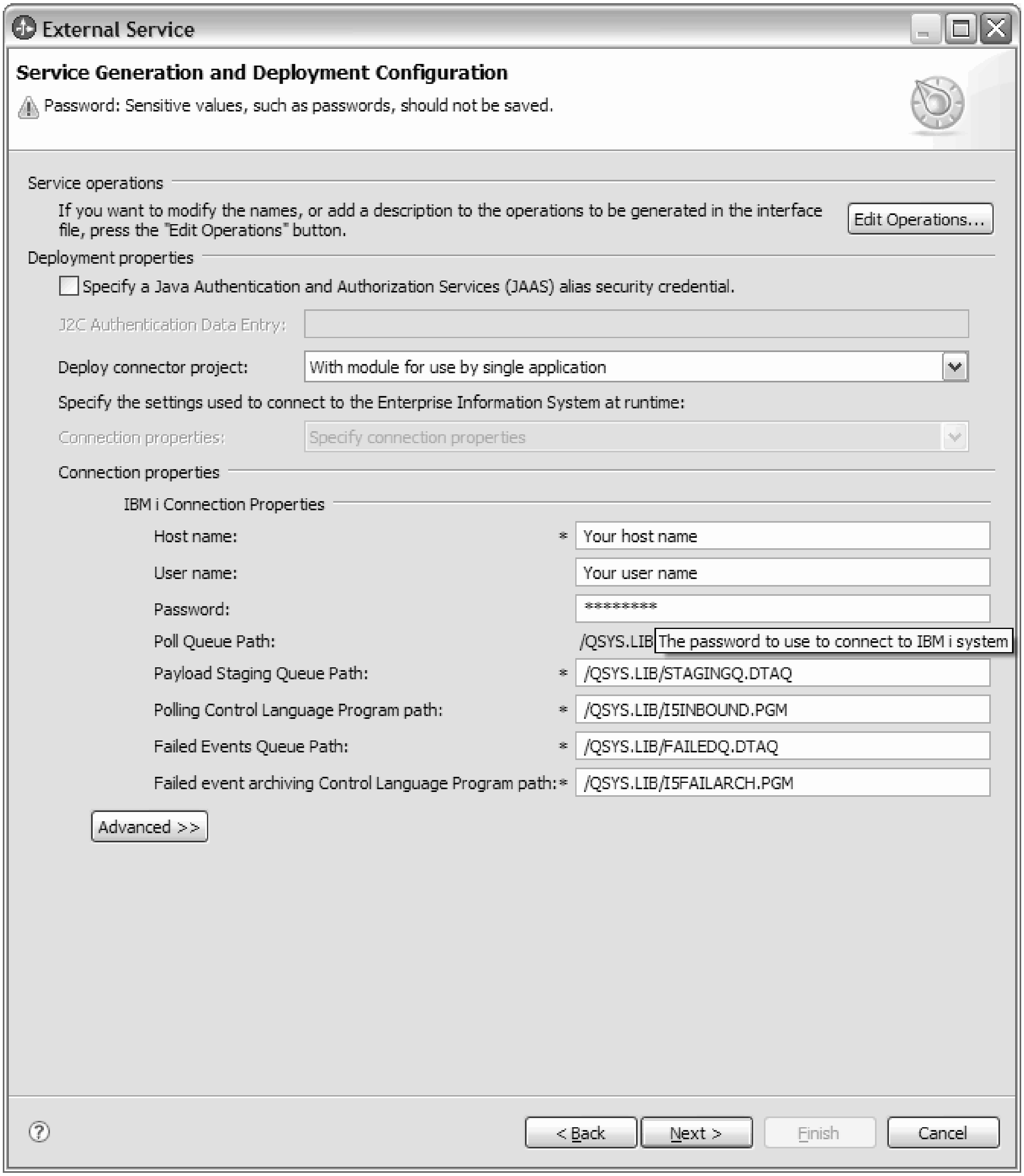
The connection properties in this task are initialized to the values that the wizard uses to connect to an IBM i system. To configure the module to use other values than those used to connect to the IBM i system, change values such as host name, user name and password here.

About this task

To generate artifacts for your module follow this procedure:

Procedure

1. In the Service Generation and Deployment Configuration window, click **Edit Operations** to review the names of or add a description for the operations for the business objects you are creating.



External Service

Service Generation and Deployment Configuration

Password: Sensitive values, such as passwords, should not be saved.

Service operations

If you want to modify the names, or add a description to the operations to be generated in the interface file, press the "Edit Operations" button.

Deployment properties

□ Specify a Java Authentication and Authorization Services (JAAS) alias security credential.

J2C Authentication Data Entry:

Deploy connector project: With module for use by single application

Specify the settings used to connect to the Enterprise Information System at runtime:

Connection properties: Specify connection properties

Connection properties

IBM i Connection Properties

Host name: * Your host name

User name: Your user name

Password: *********

Poll Queue Path: /QSYS.LIB

The password to use to connect to IBM i system

Payload Staging Queue Path: * /QSYS.LIB/STAGINGQ.DTAQ

Polling Control Language Program path: * /QSYS.LIB/I5INBOUND.PGM

Failed Events Queue Path: * /QSYS.LIB/FAILEDQ.DTAQ

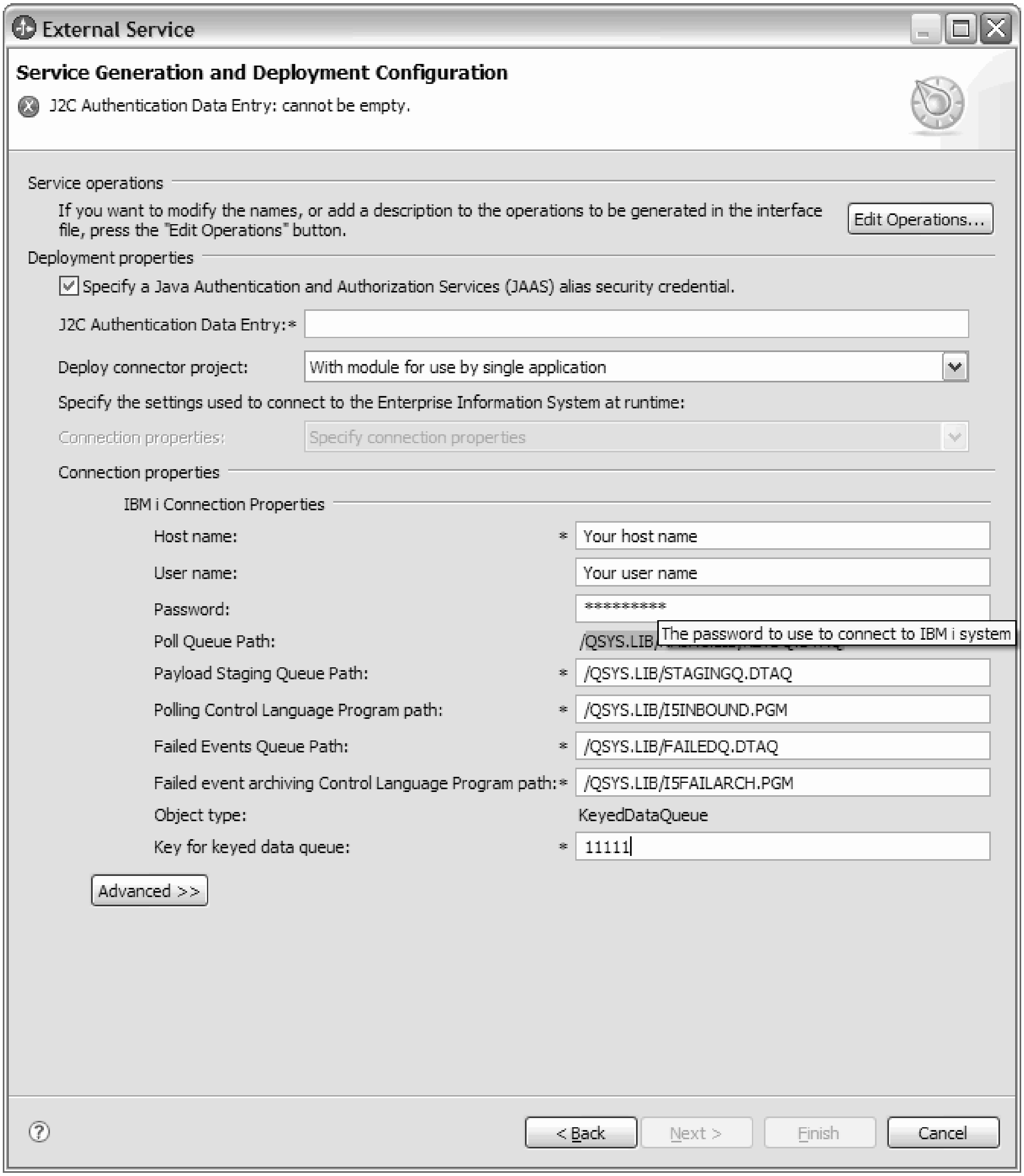
Failed event archiving Control Language Program path: * /QSYS.LIB/I5FAILARCH.PGM

Figure 15. Service Generation and Deployment Configuration window

Note: The payload staging queue path must be different for each adapter and no two adapters can have the same keyed queue as PayloadStagingQueue. Similarly, the failed events queue path must be different for each adapter and no two adapters can have the same keyed queue as FailedEventsQueue.

2. Specify how you want the adapter to connect to IBM i system at run time for simple data queues such as FIFO and LIFO, as shown in the preceding figure. Poll queue path field is disabled for simple data queue polling.

3. For a keyed queue - For example, as shown in Figure 16 on page 42, specify how you want the adapter to connect to IBM i system at run time. Type the host name, payload staging queue path, polling control language program path, failed events queue path, failed event archiving control language program path, and the key for keyed data queue values. While creating the inbound module for keyed data queue, ensure that the key value entered to be of length equivalent to the actual key length. This value is specified during creation of keyed data queue on the IBM i system. If the key value entered in enterprise metadata discovery (EMD) is either lesser or more than the maximum key length then the inbound module will not poll any message from the keyed queue.



External Service

Service Generation and Deployment Configuration

J2C Authentication Data Entry: cannot be empty.

Service operations

If you want to modify the names, or add a description to the operations to be generated in the interface file, press the "Edit Operations" button.

Deployment properties

☑ Specify a Java Authentication and Authorization Services (JAAS) alias security credential.

J2C Authentication Data Entry:*

Deploy connector project: With module for use by single application

Specify the settings used to connect to the Enterprise Information System at runtime:

Connection properties: Specify connection properties

Connection properties

IBM i Connection Properties

Host name:	*	Your host name
User name:		Your user name
Password:		**********
Poll Queue Path:		/QSYS.LIB/
Payload Staging Queue Path:	*	/QSYS.LIB/STAGINGQ.DTAQ
Polling Control Language Program path:	*	/QSYS.LIB/I5INBOUND.PGM
Failed Events Queue Path:	*	/QSYS.LIB/FAILEDQ.DTAQ
Failed event archiving Control Language Program path:	*	/QSYS.LIB/I5FAILARCH.PGM
Object type:		KeyedDataQueue
Key for keyed data queue:	*	11111

The password to use to connect to IBM i system

Figure 16. Service Generation and Deployment Configuration

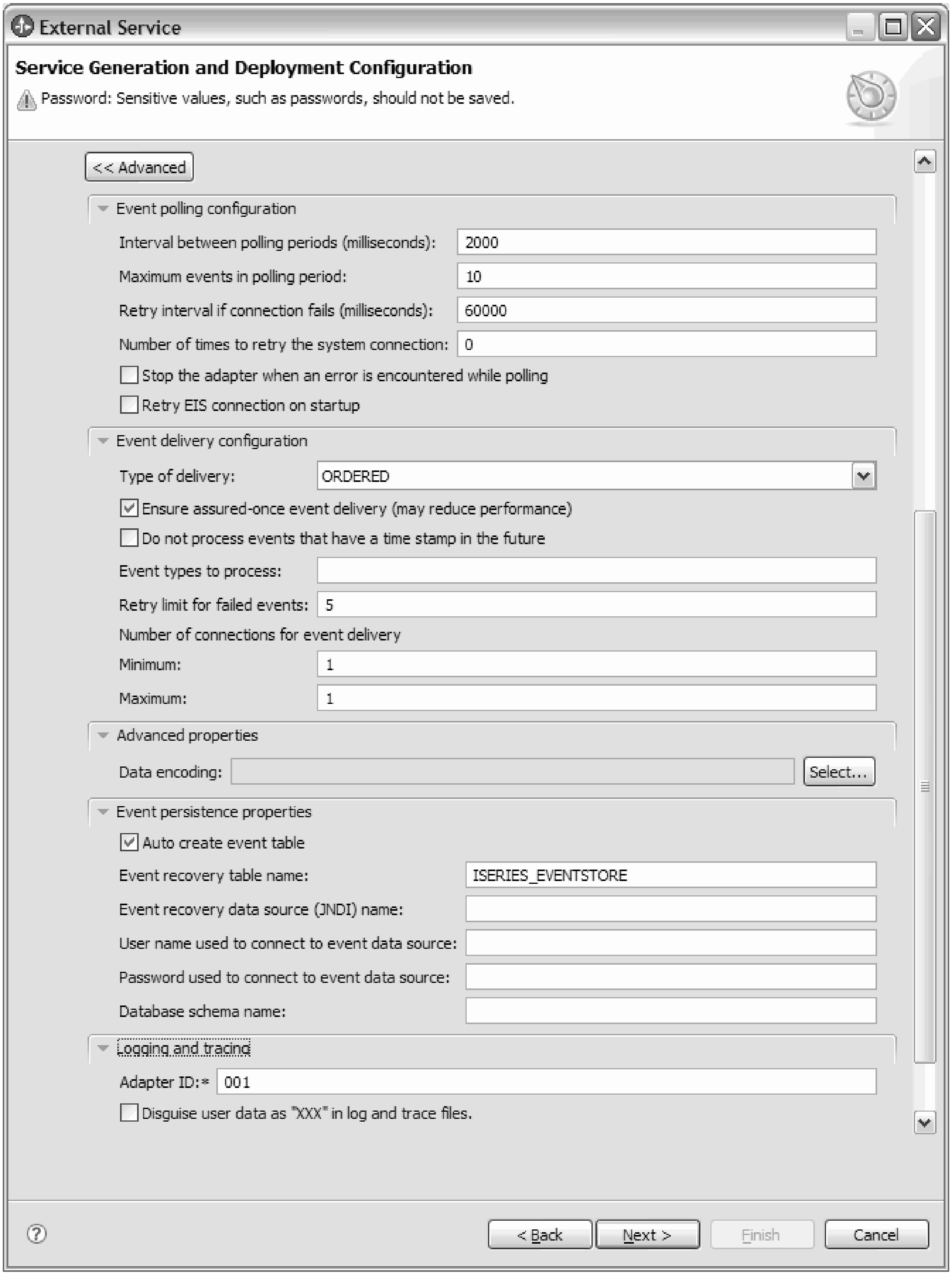
4. To use a J2C authentication alias, specify whether you want the adapter to use a **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** and type the name of the alias in **J2C Authentication Data Entry** field. You can either specify an existing authentication alias or create one at any time before deploying the module. The name is case-sensitive and includes the node name.

5. Optional: To use the user name and password specified in an existing Java Naming and Directory Interface (JNDI) data source on the server:
   a. Clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
   b. Click **Advanced**.
   c. Expand **Advanced connection configuration**.
   d. In **Datasource JNDI name** field, type the name of an existing JNDI data source.
6. Optional: To specify the database user name and password to be saved in the adapter properties:
   a. Clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
   b. Click **Advanced**.
   c. Expand **Advanced connection configuration**.
   d. In **Database system connection information** field, type a user name and password in the **User name** and **Password** fields.

      Note: When you specify the password here, it is saved as clear text in an adapter property, which unauthorized users might be able to see.
7. In **Deploy connector project**, specify whether to include the adapter files in the module. Choose one of the following values:
   - **With module for use by single application**. With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
   - **On server for use by multiple applications**. If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
8. If you selected **On server for use by multiple adapters** in the previous step, specify how you want to set the connection properties at runtime.
   - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same IBM i system using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source:
     a. In **Connection properties**, select **Use predefined connection properties**.
     b. In **JNDI Lookup Name**, type the name of the JNDI data source for an existing managed connection factory or activation specification.
     c. Click **Next**.

- If this is the first application that connects to the database with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.

9. If you selected **On server for use by multiple adapters** in the previous step, specify how you want to set the connection properties at runtime.
   - If you manually created and configured a managed connection factory or activation specification on the server or if you have already deployed an application that connects to the same IBM i server using the same managed connection factory or activation specification properties, you can reuse the managed connection factory or activation specification by specifying the name of its Java Naming and Directory Interface (JNDI) data source:
     a. In **Connection properties**, select **Use predefined connection properties**.
     b. In **JNDI Lookup Name**, type the name of the JNDI data source for an existing managed connection factory or activation specification.
     c. Click **Next**.
   - If this is the first application that connects to the database with a specific user name and password, or if you want to administer the user name and password separately from other applications, select **Specify connection properties**.
10. Optional: Review and if necessary, change the values of the required connection properties. The fields are initialized with the connection information you specified when you started the wizard. You can change the values to specify a different user name and password at run time. You can also connect to an alternate IBM i system, although the schema names must be the same in both databases. The format of the connection properties are database-specific. For more information about the properties, see the reference topic devoted to activation specification properties in this documentation.
11. For Last-in first-out (LIFO) or First-in first-out (FIFO) queue - for example, as shown in the following figure, specify advanced properties by clicking **Advanced**. Expand each of the advanced sections to review the properties. The following figure shows the advanced property sections in the Service Generation and Deployment Configuration window.



Service Generation and Deployment Configuration

Password: Sensitive values, such as passwords, should not be saved.

Event polling configuration

Interval between polling periods (milliseconds): 2000

Maximum events in polling period: 10

Retry interval if connection fails (milliseconds): 60000

Number of times to retry the system connection: 0

□ Stop the adapter when an error is encountered while polling

□ Retry EIS connection on startup

Event delivery configuration

Type of delivery: ORDERED

☑ Ensure assured-once event delivery (may reduce performance)

□ Do not process events that have a time stamp in the future

Event types to process:

Retry limit for failed events: 5

Number of connections for event delivery

Minimum: 1

Maximum: 1

Advanced properties

Data encoding:

Event persistence properties

☑ Auto create event table

Event recovery table name: ISERIES_EVENTSTORE

Event recovery data source (JNDI) name:

User name used to connect to event data source:

Password used to connect to event data source:

Database schema name:

Logging and tracing

Adapter ID:* 001

□ Disguise user data as "XXX" in log and trace files.

Figure 17. Service generation and deployment configuration advanced window

- By selecting the **Event polling configuration** property, you can set the event polling fields as follows:

a. In the **Interval between polling periods** field, type the number of milliseconds that the adapter waits between polling periods.
b. In the**Maximum events in polling period** field, type the number of events to deliver in each polling period.
c. In the **Retry interval if connection fails** field, type the number of milliseconds to wait before trying to connect after a connection failure during polling.
d. In the**Number of times to retry the system connection** field, type the number of times to retry the connection before reporting a polling error.
e. If you want the adapter to stop when polling errors occur, select **Stop the adapter when an error is encountered while polling**. If you do not select this option, the adapter logs an exception but continues to run.
f. Optional: You can select **Retry EIS connection on startup** if you want the adapter to continue trying to connect to a system to which it could not connect when starting.

- By selecting the **Event delivery configuration** property, you can set the event delivery fields as follows;

   a. In the**Type of delivery** field, select the delivery method.
   b. If you want to ensure that events are delivered only once and to only one export, select **Ensure once-only delivery**. Although this option might reduce performance, it prevents duplicate or missing events.
   c. By default, the adapter processes all events that it finds when it polls. If you do not want it to process events that have timestamps later than the current time, select **Do not process events that have a timestamp in the future**.
   d. In the **Event types to process** field, type a comma-separated list of the business objects for which you want events delivered. Leave this field blank to receive events for all business object types.

      For example, if you want to receive events only when the Customer and Order tables, but not other tables, are changed in the database, set this field to `Customer,Order`.
   e. Under the **Number of connections for event delivery** field, specify the minimum and maximum number of connections to use to deliver events.
- By selecting the **Advanced properties**, you can set select data encoding value. This value indicates the type of character encoding the adapter will use during data transformation. For more information about the encoding property, see the reference topic devoted to IBM i business object properties in this documentation.
- By selecting the **Event persistence properties**, you can make sure that events are delivered once, and only once, to the export in the case of a failure. You can set the event persistence fields as follows:

   a. Select the **Auto create event table** check box, to create the event persistence table automatically.
   b. In the **Event recovery table name** field, enter the name of the table used by the adapter for event persistence processing.
   c. In the **Event recovery data source (JNDI) name** field, enter the JNDI name of the data source used by event persistence processing to obtain the JDBC database connection. The data source must be created in WebSphere Process Server.

d. In the **User name used to connect to event data source** field, enter your database user name.

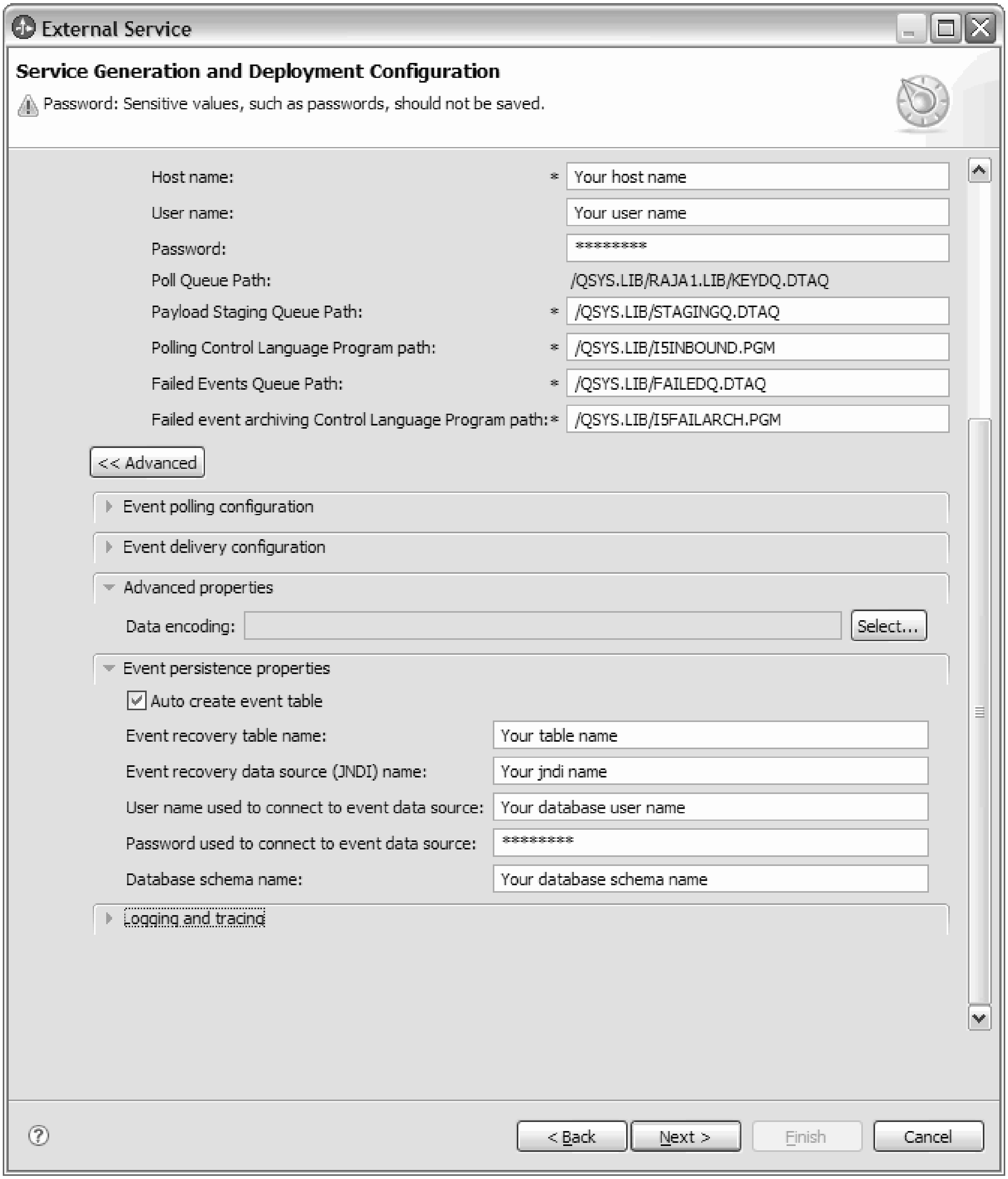
e. In the**Password used to connect to event data source** field, enter your password.

f. In the**Database schema name** field, enter the name of the database used by event persistence processing.

- By selecting the**Logging and tracing** property, you can set information about the runtime state of the adapter.

  a. If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.

12. For a keyed queue - For example, as shown in following figure, specify advanced properties by clicking **Advanced**. Expand each of the advanced sections to review the properties. The following figure shows the advanced property sections in the Service Generation and Deployment Configuration window for a keyed queue.



Service Generation and Deployment Configuration

⚠ Password: Sensitive values, such as passwords, should not be saved.

Host name: * Your host name

User name: Your user name

Password: *********

Poll Queue Path: /QSYS.LIB/RAJA1.LIB/KEYDQ.DTAQ

Payload Staging Queue Path: * /QSYS.LIB/STAGINGQ.DTAQ

Polling Control Language Program path: * /QSYS.LIB/I5INBOUND.PGM

Failed Events Queue Path: * /QSYS.LIB/FAILEDQ.DTAQ

Failed event archiving Control Language Program path: * /QSYS.LIB/I5FAILARCH.PGM

Event polling configuration

Event delivery configuration

Advanced properties

Data encoding:

Event persistence properties

☑ Auto create event table

Event recovery table name: Your table name

Event recovery data source (JNDI) name: Your jndi name

User name used to connect to event data source: Your database user name

Password used to connect to event data source: *********

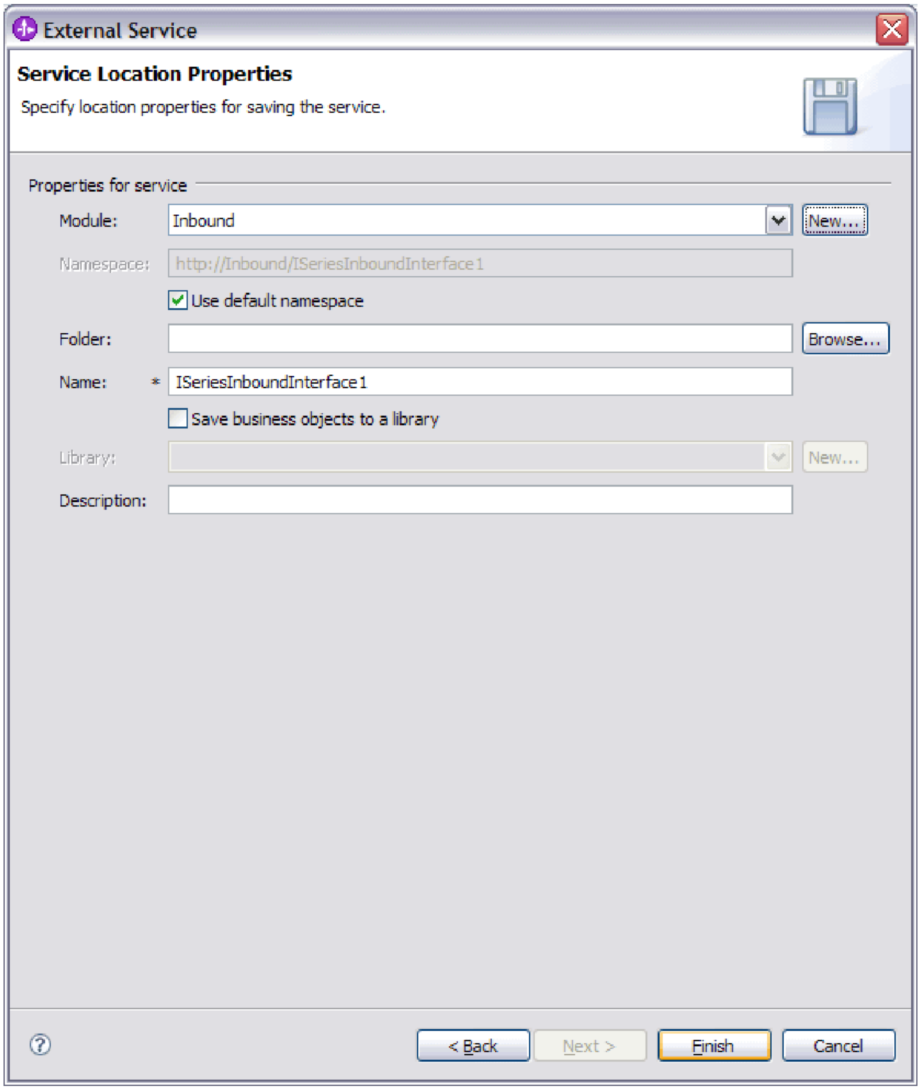
Database schema name: Your database schema name

Logging and tracing

Figure 18. Service generation and deployment configuration window

For more details of the Advanced properties such as Event polling configuration, Event delivery configuration, EventPersistenceProperties and Logging and tracing refer to the respective section in step 11.

13. Click **Next**. The Service Location Properties window is displayed.


External Service

Service Location Properties

Specify location properties for saving the service.

Properties for service

Module: Inbound

Namespace: http://Inbound/ISeriesInboundInterface1

☑ Use default namespace

Folder:

Name: * ISeriesInboundInterface1

□ Save business objects to a library

Library:

Description:

Figure 19. Service Location Properties window

14. In the Service Location Properties window, specify the name of the module you want to create. This can be a new or existing module.
    - If the desired module name appears in the **Module** list, select its name.

      Important: If the module contains an interface or business object with the same name as any you are now configuring, the original interface or business object in the module is replaced by the new version.
    - Otherwise, create a new module:
      a. Click **New**.
      b. In theIntegration Project window, select **Create a module project** and click **Next**.
      c. In the Module window, type a name for the module. For example, ISeriesInboundModule.
      d. If you want the service description files (the .export and .wsdl files) to be located in the default folder in the module, leave **Use default**

location check box selected. If you want to specify a different folder in the module, clear the option and then click **Browse** to specify a different folder in **Location**.

   e. If you want the module to be automatically opened in the assembly diagram in WebSphere Integration Developer when the wizard closes, select **Open module assembly diagram**. Otherwise, clear this option.
   f. Click **Finish** to create the new module.

15. Optional: Specify the namespace you want to use for your business objects.
    - If you want the business objects in the module to use the default namespace, leave **Use default namespace** selected.
    - To specify a different namespace, clear the option and type a different value in **Namespace**.
16. Optional: Specify the folder within the new module where the service description will be saved. In **Folder**, type the folder name or browse to an existing folder. If you do not specify a folder name, the artifacts (the export, XSD, and WSDL files) are stored in the root folder of the module, that is, the folder with the module name.
17. In **Name**, accept the default name of the import or type a different name.
18. Optionally, if you want to save the business objects in a library where they can be used by other modules, select **Save business objects to a library** and specify the location of the library in **Library**.
19. Optionally, in **Description**, type a descriptive comment about the module.
20. When you are finished setting properties, click **Finish**.
21. If the Model Changed window is displayed, click **Yes**.

Results

The wizard exits. The module is created in the project and artifacts are generated.

What to do next

In some instances, you might need to use the assembly editor to complete the configuration. Then you can test or deploy your module.

Chapter 5. Changing interaction specification properties using the assembly editor

To change interaction specification properties for your adapter module after generating the service, use the assembly editor in WebSphere Integration Developer.

Before you begin

You must have used the external service wizard to generate a service for the adapter.

About this task

You might want to change interaction specification properties after you have generated a service for the adapter. Interaction specification properties, which are optional, are set at the method level, for a specific operation on a specific business object. The values you specify will appear as defaults in all parent business objects generated by the external service wizard. You can change these properties before you export the EAR file. You cannot change these properties after you deploy the application.

To change the interaction specification properties, use the following procedure.

Procedure

1. From the Business Integration perspective of WebSphere Integration Developer, expand the module name.
2. Expand **Assembly Diagram** and double-click the interface.
3. Click the interface in the assembly editor. (It shows the module properties if you don't do the extra click.)
4. Click the **Properties** tab. (You can also right-click the interface in the diagram and click **Show in Properties**.)
5. Under **Binding**, click **Method bindings**. The methods for the interface are displayed, one for each combination of business object and operation.
6. Select the method whose interaction specification property you want to change.

Results

The interaction specification properties associated with your adapter module are changed.

What to do next

Deploy the module.

Chapter 6. Deploying the module

Deploy a module to place the files that make up your module and adapter into an operational environment for production or testing. In WebSphere Integration Developer, the integrated test environment features runtime support for WebSphere Process Server or WebSphere Enterprise Service Bus, or both, depending on the test environment profiles that you selected during installation.

Deployment environments

There are test and production environments into which you can deploy modules and adapters.

In WebSphere Integration Developer, you can deploy your modules to one or more servers in the test environment. This is typically the most common practice for running and testing business integration modules. However, you can also export modules for server deployment on WebSphere Process Server or WebSphere Enterprise Service Bus as EAR files using the administrative console or command-line tools.

Deploying the module for testing

In WebSphere Integration Developer, you can deploy a module that includes an embedded adapter to the test environment and work with server tools that enable you to perform such tasks as editing server configurations, starting and stopping servers and testing the module code for errors. The testing is generally performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

About this task

Generating and wiring a target component for testing inbound processing

Before deploying to the test environment a module that includes an adapter for inbound processing, you must first generate and wire a target component. This target component serves as the *destination* to which the adapter sends events.

Before you begin

You must have generated an export module, using the external service wizard.

About this task

Generating and wiring a target component for inbound processing is required in a testing environment only. It is not necessary when deploying the adapter in a production environment.

The target component receives events. You *wire* the export to the target component (connecting the two components) using the assembly editor in WebSphere Integration Developer. The adapter uses the wire to pass event data (from the export to the target component).

Procedure

1. Create the target component
   a. From the Business Integration perspective of WebSphere Integration Developer, expand **Assembly Diagram** and double-click the export component. If you did not change the default value, the name of the export component is the name of your adapter + **InboundInterface**.

      An interface specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions. The **InboundInterface** contains the operations required by the adapter to support inbound processing and is created when you run the external service wizard.
   b. Create a new component by expanding **Components**, selecting **Untyped Component**, and dragging the component to the Assembly Diagram.

      The cursor changes to the placement icon.
   c. Click the component to have it displayed in the Assembly Diagram.
2. Wire the components.
   a. Click and drag the export component to the new component. This draws a wire from the export component to the new component, as shown in the following figure:
   b. Save the assembly diagram. Click **File** → **Save**
3. Generate an implementation for the new component.
   a. Right-click on the new component and select **Generate implementation**.

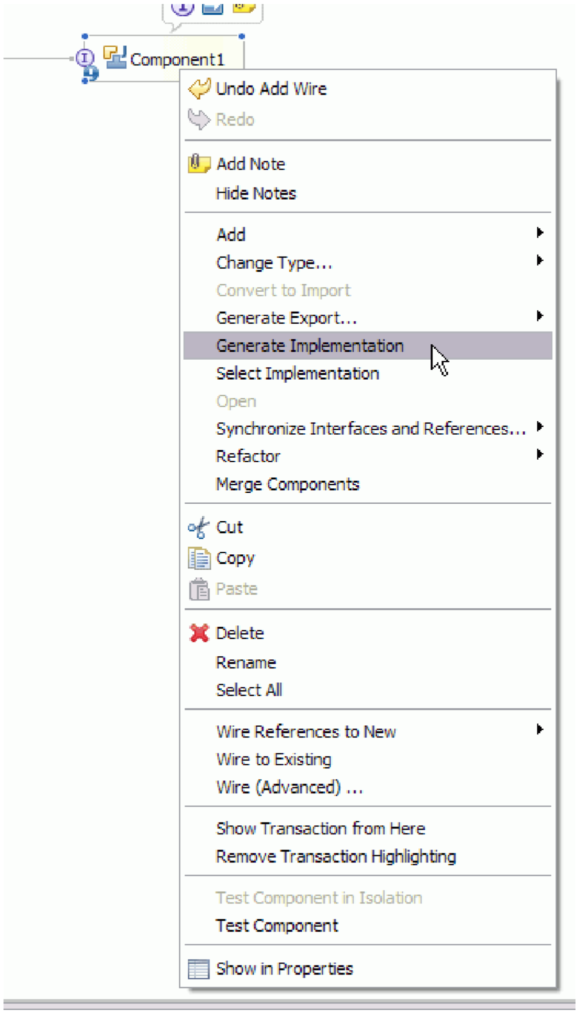


Figure 20. Generating a Java implementation

b. Select **(default package)** and click **OK**. This creates an endpoint for the inbound module.

   The Java implementation is displayed in a separate tab.

c. **Optional:** Add print statements to print the data object received at the endpoint for each of the endpoint methods.

d. Click **File** → **Save** to save the changes.

What to do next

Continue deploying the module for testing.

Adding the module to the server

In WebSphere Integration Developer, you can add modules to one or more servers in the test environment.

Before you begin

If the module you are testing uses an adapter to perform inbound processing, you need to generate and wire a *target component* to which the adapter will send events.

About this task

In order to test your module and its use of the adapter, you need to add the module to the server.

Procedure

1. *Conditional:* If there are no servers in the **Servers view**, add and define a new server by performing the following steps:
   a. Place your cursor in the **Servers view**, right click and select **New** → **server**
   b. From the Define a New Server window, select the server type.
   c. Configure server's settings.
   d. Click **Finish** to publish the server.
2. Add the module to the server
   a. Switch to the servers view. In WebSphere Integration Developer, select **Windows** → **Show View** → **Servers**
   a. Start the server. In the Servers tab in the lower-right pane of the WebSphere Integration Developer screen, right-click on the server, and then select **Start**.
3. When the server status is *Started*, right-click on the server, and select **Add and remove projects**.
4. In the Add and Remove Projects screen, select your project and click **Add**. The project moves from the **Available projects** list to the **Configured projects** list.
5. Click **Finish**. This deploys the module on the server.

   The Console tab in the lower-right pane displays a log while the module is being added to the server.

What to do next

Test the functionality of your module and the adapter.

Testing the module for outbound processing using the test client

Test the assembled module and adapter for outbound processing using the WebSphere Integration Developer integration test client.

Before you begin

You need to add the module to the server first.

About this task

Testing a module is generally performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

Procedure

1. Select the module you want to test, right-click on it, and select **Test** → **Test Module**.
2. For information on testing a module using the test client, see the *Testing modules and components* topic in the WebSphere Integration Developer information center.

What to do next

If you are satisfied with the results of testing your module and adapter, you can deploy the module and adapter to the production environment.

Deploying the module for production

Deploying a module created with the external service wizard to WebSphere Process Server or WebSphere Enterprise Service Bus in a production environment is a two-step process. First, you export the module in WebSphere Integration Developer as an enterprise archive (EAR) file. Second, you deploy the EAR file using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

About this task

Installing the RAR file (for modules using stand-alone adapters only)

If you chose not to embed the adapter with your module, but instead choose to make the adapter available to all deployed applications in the server instance, you will need to install the adapter in the form of a RAR file to the application server. A RAR file is a Java archive (JAR) file that is used to package a resource adapter for the Java 2 Connector (J2C) architecture.

Before you begin

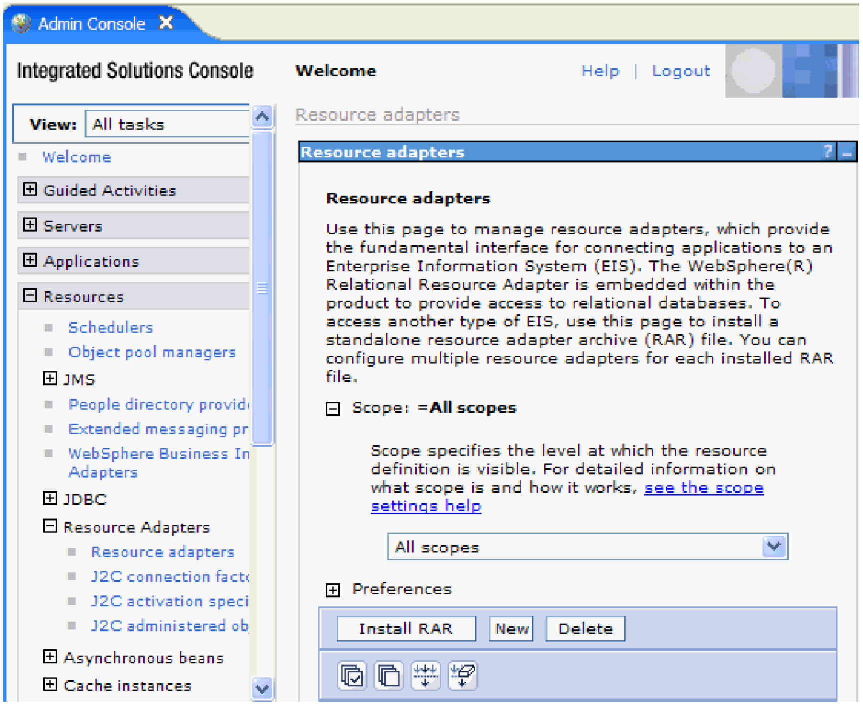
You must have set **Deploy connector project** to **On server for use by multiple adapters** in the Service Generation and Deployment Configuration window of the external service wizard.

About this task

Installing the adapter in the form of a RAR file results in the adapter being available to all J2EE application components running in the server runtime.

Procedure

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **Install RAR**.



Resource adapters

Use this page to manage resource adapters, which provide the fundamental interface for connecting applications to an Enterprise Information System (EIS). The WebSphere(R) Relational Resource Adapter is embedded within the product to provide access to relational databases. To access another type of EIS, use this page to install a standalone resource adapter archive (RAR) file. You can configure multiple resource adapters for each installed RAR file.

Scope: =**All scopes**

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help

All scopes

Figure 21. The Install RAR button on the Resource adapters page

4. In the Install RAR file page, click **Browse** and navigate to the RAR file for your adapter.

   The RAR files are typically installed in the following path: *WID_installation_directory*/ResourceAdapters/*adapter_name*/deploy/*adapter*.rar
5. Click **Next**.
6. In the Resource adapters page, optionally change the name of the adapter and add a description.
7. Click **OK**.
8. Click **Save** in the **Messages** box at the top of the page.

What to do next

The next step is to export the module as an EAR file that you can deploy on the server.

Exporting the module as an EAR file

Using WebSphere Integration Developer, export your module as an EAR file. By creating an EAR file, you capture all of the contents of your module in a format that can be easily deployed to WebSphere Process Server or WebSphere Enterprise Service Bus.

Before you begin

Before you can export a module as an EAR file, you must have created a module to communicate with your service. The module should be displayed in the WebSphere Integration Developer Business Integration perspective.

About this task

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

Procedure

1. Right-click the module and select **Export**.
2. In the Select window, expand **Java EE**.
3. Select **EAR file** and click **Next**.
4. Optional: Select the correct EAR application. The EAR application is named after your module, but with "App" added to the end of the name.
5. **Browse** for the folder on the local file system where the EAR file will be placed.
6. Optionally, if you want to export the source files, select **Export source files**. This option is provided in case you want to export the source files in addition to the EAR file. Source files include files associated with Java components, data maps, and so on.
7. To overwrite an existing file, click **Overwrite an existing file**.
8. Click **Finish**.

Results

The contents of the module are exported as an EAR file.

Install the module in the administrative console. This deploys the module to WebSphere Process Server or WebSphere Enterprise Service Bus.

Installing the EAR file

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

Before you begin

You must have exported your module as an EAR file before you can install it on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

To install the EAR file, perform the following procedure. For more information on clustering adapter module applications, see the http://www.ibm.com/software/webservers/appserv/was/library/.

Procedure

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

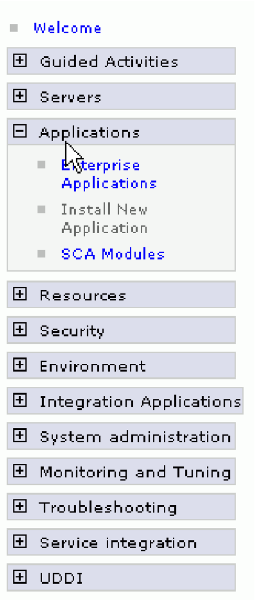


Figure 22. Preparing for the application installation window

3. Click **Browse** to locate your EAR file and click **Next**. The EAR file name is the name of the module followed by ″App.″
4. Optional: If you are deploying to a clustered environment, complete the following steps.
   a. On the **Step 2: Mapping modules to servers** window, select the module.
   b. Select the name of the server cluster.
   c. Click **Apply**.
5. Click **Next** to open the Summary. Verify that all settings are correct and click **Finish**.
6. Optional: If you are using an authentication alias, complete the following steps:
   a. Expand **Security** and select **Business Integration Authentication Aliases**.
   b. Select the authentication alias that you want to configure. You must have administrator or operator authority to make changes to authentication alias configurations.
   c. Optional: If it is not already filled in, type the **User name**.
   d. If it is not already filled in, type the **Password**.
   e. If it is not already filled in, type the password again in the **Confirm Password** field.
   f. Click **OK**.

Results

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

What to do next

If you want to set or reset any properties or you would like to cluster adapter project applications, make those changes using the administrative console before configuring troubleshooting tools.

Chapter 7. Administering the adapter module

When you are running the adapter in a stand-alone deployment, use the administrative console of the server to start, stop, monitor, and troubleshoot the adapter module. In an application that uses an embedded adapter, the adapter module starts or stops when the application is started or stopped.

Changing configuration properties for embedded adapters

To change configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment.

Setting resource adapter properties for embedded adapters

To set resource adapter properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

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

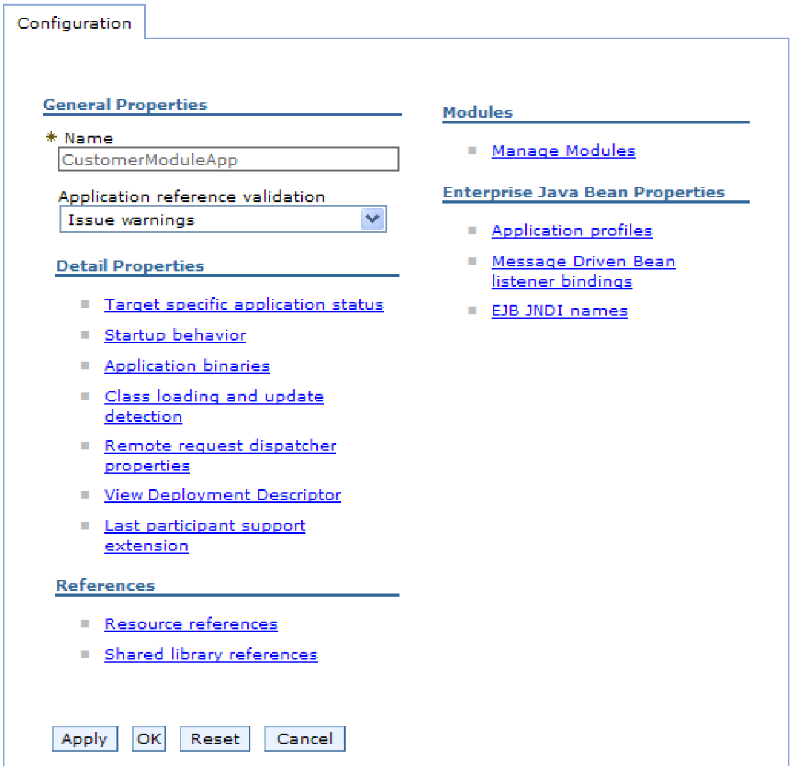
About this task

Custom properties are default configuration properties shared by all WebSphere adapters.

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

Procedure

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change. The Configuration page is displayed.


Configuration
General Properties
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Remote request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Resource references
Shared library references
Modules
Manage Modules
Enterprise Java Bean Properties
Application profiles
Message Driven Bean listener bindings
EJB JNDI names
Apply
OK
Reset
Cancel

* Name: CustomerModuleApp

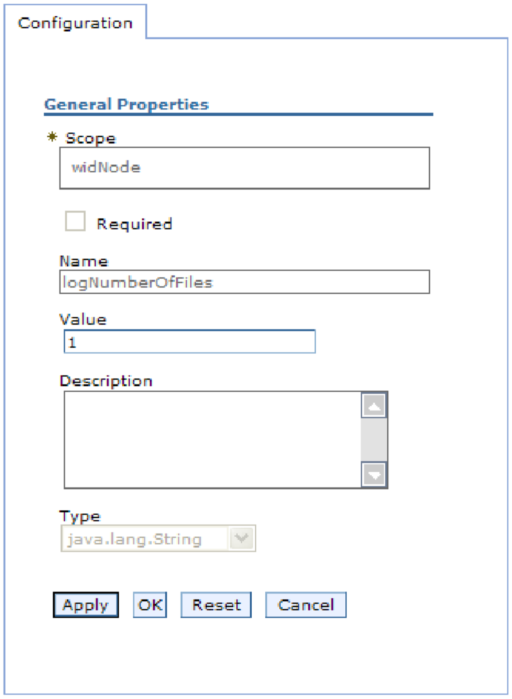
Application reference validation: Issue warnings

Figure 23. The Manage Modules selection in the Configuration tab

4. Under **Modules**, click **Manage Modules**.
5. Click **IBM WebSphere Adapter for IBM i**.
6. From the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **Custom properties**.
8. For each property you want to change, perform the following steps.

   Note: See "Resource adapter properties" on page 91 for more information about these properties.

   a. Click the name of the property. The **Configuration** page for the selected property is displayed. For example, if you click the **logNumberOfFiles** property, you see the page shown in the figure below.



Configuration

General Properties

* Scope
widNode

□ Required

Name
logNumberOfFiles

Value
1

Description

Type
java.lang.String

Figure 24. The Configuration tab for the logNumberOfFiles property

b. Change the contents of the **Value** field or type a value, if the field is empty.

   You can change the number in the **Value** field and add a description of the property.

c. Click **OK**.

9. Click the **Save** link in the **Messages** box at the top of the window.

Results

The resource adapter properties associated with your adapter module are changed.

Setting managed (J2C) connection factory properties for embedded adapters

To set managed connection factory properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

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

About this task

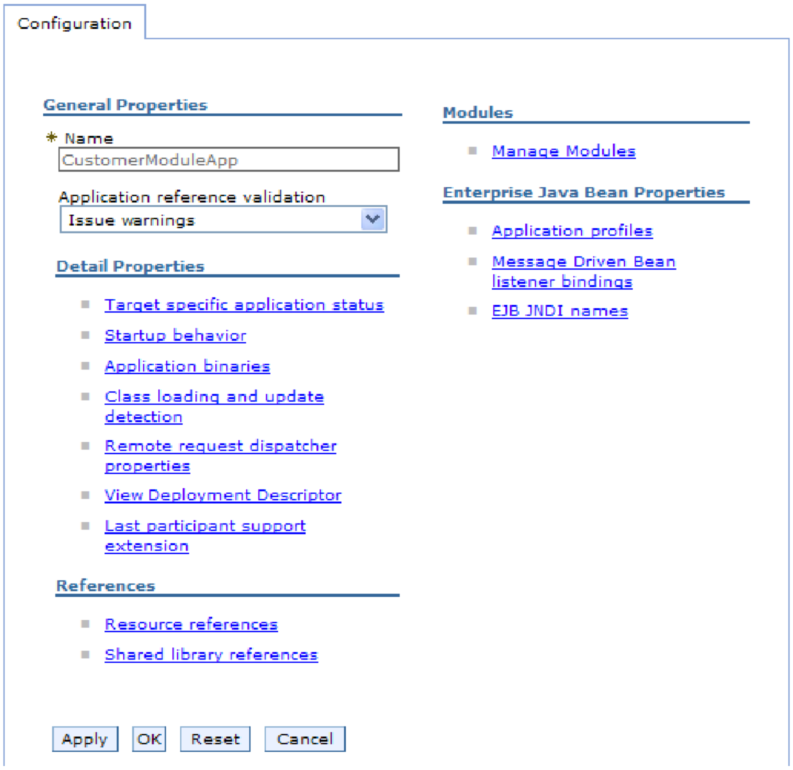
You use managed connection factory properties to configure the target IBM i system instance.

Note: In the administrative console, the properties are referred to as ″J2C connection factory properties.″

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

Procedure

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. In the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change.
4. Under **Modules**, click **Manage Modules**.



* Name: CustomerModuleApp

Application reference validation: Issue warnings

Configuration
General Properties
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Remote request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Resource references
Shared library references
Modules
Manage Modules
Enterprise Java Bean Properties
Application profiles
Message Driven Bean listener bindings
EJB JNDI names
Apply
OK
Reset
Cancel

Figure 25. The Manage Modules selection in the Configuration tab

5. Click **IBM WebSphere Adapter for IBM i**.
6. In the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
8. Click the name of the connection factory associated with your adapter module.
9. In the **Additional Properties** list, click **Custom properties**.

   Custom properties are those J2C connection factory properties that are unique to Adapter for IBM i. 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.

    Note: See "Managed connection factory properties" on page 93 for more information about these properties.

    a. Click the name of the property.

   b. Change the contents of the **Value** field or type a value, if the field is empty.
   c. Click **OK**.
11. Click the **Save** link in the **Messages** box at the top of the window.

Results

The managed connection factory properties associated with your adapter module are changed.

Setting activation specification properties for embedded adapters

To set activation specification properties for your adapter after it has been deployed as part of a module, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

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

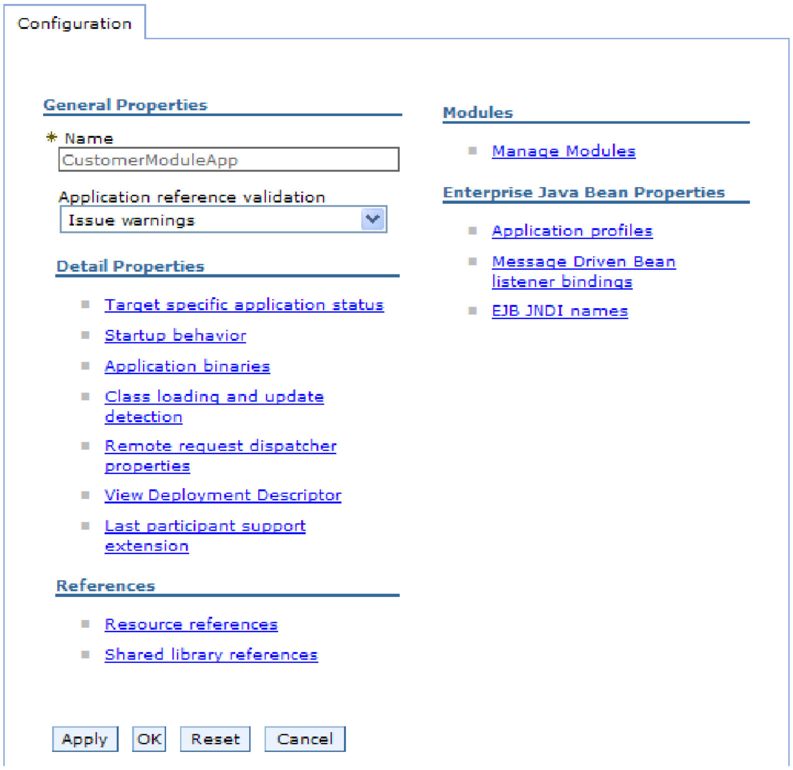
About this task

You use activation specification properties to configure the endpoint for inbound processing.

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

Procedure

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change.
4. Under **Modules**, click **Manage Modules**.


Configuration
General Properties
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Remote request dispatcher properties
View Deployment Descriptor
Last participant support extension
References
Resource references
Shared library references
Modules
Manage Modules
Enterprise Java Bean Properties
Application profiles
Message Driven Bean listener bindings
EJB JNDI names
Apply
OK
Reset
Cancel

* Name: CustomerModuleApp

Application reference validation: Issue warnings

Figure 26. The Manage Modules selection in the Configuration tab

5. Click **IBM WebSphere Adapter for IBM i**.
6. From the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **J2C activation specifications**.
8. Click the name of the activation specification associated with the adapter module.
9. From the **Additional Properties** list, click **J2C activation specification custom properties**.
10. For each property you want to change, perform the following steps.

    Note: See "Activation specification properties" on page 103 for more information about these properties.

    a. Click the name of the property.
    b. Change the contents of the **Value** field or type a value, if the field is empty.
    c. Click **OK**.
11. Click the **Save** link in the **Messages** box at the top of the window.

Results

The activation specification properties associated with your adapter module are changed.

Changing configuration properties for stand-alone adapters

To set configuration properties after you install a stand-alone adapter, you use the administrative console of the runtime environment. You provide general information about the adapter and then set resource adapter properties (which are used for general adapter operation). If the adapter will be used for outbound operations, you create a connection factory and then set properties for it.

Setting resource adapter properties for stand-alone adapters

To set resource adapter properties for your stand-alone adapter after it has been installed on WebSphere Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

Custom properties are default configuration properties shared by all WebSphere adapters.

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

Procedure

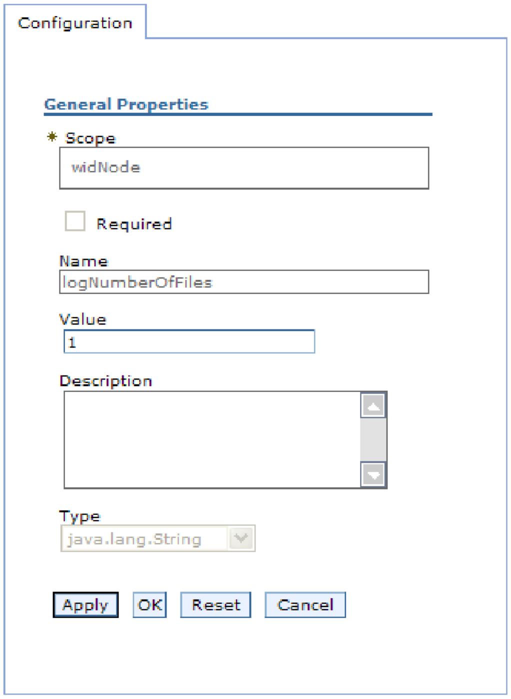
1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for IBM i**.
4. In the **Additional Properties** list, click **Custom properties**.
5. For each property you want to change, perform the following steps.

   Note: See "Resource adapter properties" on page 91 for more information about these properties.

   a. Click the name of the property.

   b. Change the contents of the **Value** field or type a value, if the field is empty.

   For example, if you click **logNumberOfFiles**, you see the following page:


Configuration

General Properties

* Scope
widNode

□ Required

Name
logNumberOfFiles

Value
1

Description

Type
java.lang.String

Figure 27. The Configuration tab for the logNumberOfFiles property

You can change the number in the **Value** field and add a description of the property.

c. Click **OK**.

6. Click **Save** in the **Messages** box at the top of the page.

Results

The resource adapter properties associated with your adapter are changed.

Setting managed (J2C) connection factory properties for stand-alone adapters

To set managed connection factory properties for your stand-alone adapter after it has been installed on WebSphere Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the property you want to configure and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

You use managed connection factory properties to configure the target IBM i system instance.

Note: In the administrative console, the properties are referred to as ″J2C connection factory properties.″

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

Procedure

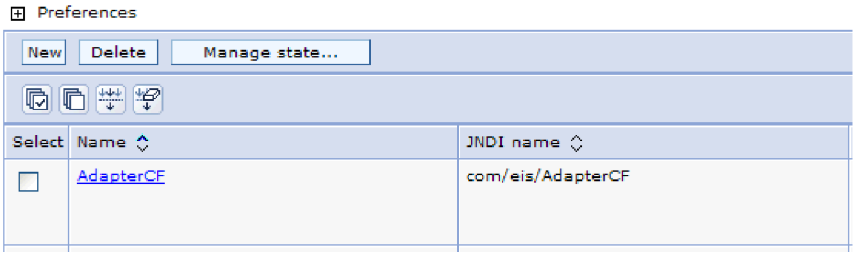
1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for IBM i**.
4. In the **Additional Properties** list, click **J2C connection factories**.
5. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

   Note: If you selected **Use predefined connection properties** when you used the external service wizard to configure the adapter module, you do not need to create a connection factory.

   If you are creating a connection factory, perform the following steps:

   a. Click **New**.
   b. In the **General Properties** section of the **Configuration** tab, type a name for the connection factory. For example, you could type `AdapterCF`.
   c. Type a value for **JNDI name**. For example, you could type `com/eis/AdapterCF`.
   d. Select an authentication alias from the **Component-managed authentication alias** list.
   e. Click **OK**.
   f. Click **Save** in the **Messages** box at the top of the page.

      The newly created connection factory is displayed.



Select	Name ⇕	JNDI name ⇕
□	AdapterCF	com/eis/AdapterCF

Figure 28. User defined connection factories for use with the resource adapter

6. In the list of connection factories, click the one you want to use.
7. In the **Additional Properties** list, click **Custom properties**.

   Custom properties are those J2C connection factory properties that are unique to Adapter for IBM i. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
8. For each property you want to change, perform the following steps.

   Note: See “Managed connection factory properties” on page 93 for more information about these properties.

   a. Click the name of the property.
   b. Change the contents of the **Value** field or type a value, if the field is empty.
   c. Click **OK**.

9. After you have finished setting properties, click **Apply**.
10. Click **Save** in the **Messages** box at the top of the window.

Results

The managed connection factory properties associated with your adapter are set.

Setting activation specification properties for stand-alone adapters

To set activation specification properties for your stand-alone adapter after it has been installed on WebSphere Process Server or WebSphere Enterprise Service Bus, use the administrative console. You select the name of the message endpoint property you want to configure, and then change or set the value.

Before you begin

Your adapter must be installed on WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

You use activation specification properties to configure the endpoint for inbound processing.

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

Procedure

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for IBM i**.
4. In the **Additional Properties** list, click **J2C activation specifications**.
5. If you are going to use an existing activation specification, skip ahead to select from an existing list of activation specifications.

   Note: If you selected **Use predefined connection properties** when you used the external service wizard to configure the adapter module, you do not need to create an activation specification.

   If you are creating an activation specification, perform the following steps:

   a. Click **New**.
   b. In the **General Properties** section of the **Configuration** tab, type a name for the activation specification. For example, you could type `AdapterAS`.
   c. Type a value for **JNDI name**. For example, you could type `com/eis/AdapterAS`.
   d. Select an authentication alias from the **Authentication alias** list.
   e. Select a message listener type.
   f. Click **OK**.
   g. Click **Save** in the **Messages** box at the top of the page.

      The newly created activation specification is displayed.
6. In the list of activation specifications, click the one you want to use.

7. In the Additional Properties list, click **J2C activation specification custom properties**.
8. For each property you want to set, perform the following steps.

   Note: See "Activation specification properties" on page 103 for more information about these properties.
   1. Click the name of the property.
   2. Change the contents of the **Value** field or type a value, if the field is empty.
   3. Click **OK**.
9. After you have finished setting properties, click **Apply**.
10. Click **Save** in the **Messages** box at the top of the page.

Results

The activation specification properties associated with your adapter are set.

Starting the application that uses the adapter

Use the administrative console of the server to start an application that uses the adapter. By default, the application starts automatically when the server starts.

About this task

Use this procedure to start the application, whether it is using an embedded or a stand-alone adapter. For an application that uses an embedded adapter, the adapter starts when the application starts. For an application that uses a stand-alone adapter, the adapter starts when the application server starts.

Procedure

1. On the administrative console, click **Applications** → **Enterprise Applications**.

   Note: The administrative console is labeled "Integrated Solutions Console".
2. Select the check box of the application that you want to start. The application name is the name of the EAR file you installed, without the .EAR file extension.
3. Click **Start**.

Results

The status of the application changes to Started, and a message stating that the application has started displays at the top of the administrative console.

Stopping the application that uses the adapter

Use the administrative console of the server to stop an application that uses the adapter. By default, the application stops automatically when the server stops.

About this task

Use this procedure to stop the application, whether it is using an embedded or a stand-alone adapter. For an application with an embedded adapter, the adapter stops when the application stops. For an application that uses a stand-alone adapter, the adapter stops when the application server stops.

Procedure

1. On the administrative console, click **Applications** → **Enterprise Applications**.

   Note: The administrative console is labeled “Integrated Solutions Console”.
2. Select the check box of the application that you want to stop. The application name is the name of the EAR file you installed, without the .EAR file extension.
3. Click **Stop**.

Results

The status of the application changes to Stopped, and a message stating that the application has stopped displays at the top of the administrative console.

Monitoring performance using Performance Monitoring Infrastructure

Performance Monitoring Infrastructure (PMI) is a feature of the administrative console that allows you to dynamically monitor the performance of components in the production environment, including the adapter for IBM i. PMI collects adapter performance data, such as average response time and total number of requests, from various components in the server and organizes the data into a tree structure. You can view the data through the Tivoli® Performance Viewer, a graphical monitoring tool that is integrated with the administrative console in WebSphere Process Server.

About this task

Before you can enable and configure PMI for your adapter, you must first set the level of tracing detail and run some events from which to gather performance data.

To learn more about how PMI can help you monitor and improve the overall performance of your adapter environment, search for PMI on the WebSphere Application Server web site: http://www.ibm.com/software/webservers/appserv/was/library/.

Configuring Performance Monitoring Infrastructure

You can configure Performance Monitoring Infrastructure (PMI) to gather adapter performance data, such as average response time and total number of requests. After you configure PMI for your adapter, you can monitor the adapter performance using Tivoli Performance viewer.

Before you begin

Before you can configure PMI for your adapter, you must first set the level of tracing detail and run some events from which to gather performance data.

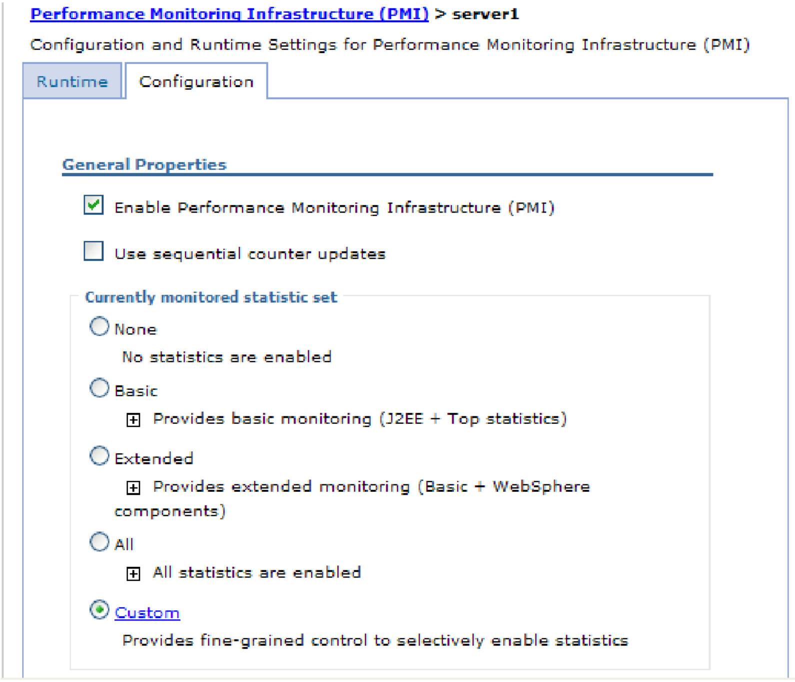
1. To enable tracing and to receive event data, the trace level must be set to either fine, finer, finest, or all. After *=info, add a colon and a string, for example:

   ```
   *=info: WBILocationMonitor.CEI.ResourceAdapter.
   *=finest: WBILocationMonitor.LOG.ResourceAdapter.*=finest:
   ```

   For instructions on setting the trace level, refer to “Enabling tracing with the Common Event Infrastructure (CEI)” on page 75.
2. Generate at least one outbound request to produce performance data that you can configure.

Procedure

1. Enable PMI for your adapter.
   a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
   b. From the list of servers, click the name of your server.
   c. Select the Configuration tab, and then select the **Enable Performance Monitoring (PMI)** check box.
   d. Select **Custom** to selectively enable or disable statistics.



Performance Monitoring Infrastructure (PMI) > **server1**

Configuration and Runtime Settings for Performance Monitoring Infrastructure (PMI)

General Properties

☑ Enable Performance Monitoring Infrastructure (PMI)

□ Use sequential counter updates

Currently monitored statistic set

○ None
No statistics are enabled

○ Basic
⊞ Provides basic monitoring (J2EE + Top statistics)

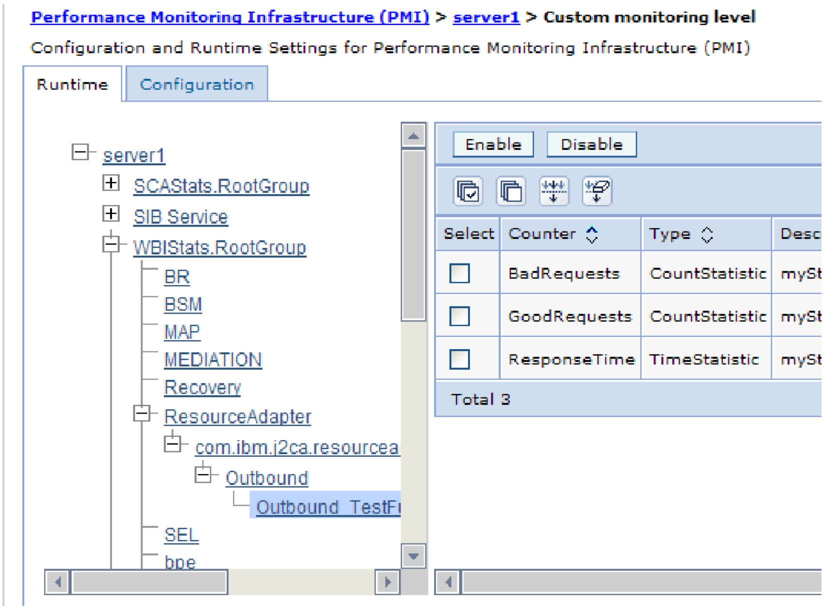
○ Extended
⊞ Provides extended monitoring (Basic + WebSphere components)

○ All
⊞ All statistics are enabled

● Custom
Provides fine-grained control to selectively enable statistics

Figure 29. Enabling Performance Monitoring Infrastructure

   e. Click **Apply** or **OK**.
   f. Click **Save**. PMI is now enabled.
2. Configure PMI for your adapter.
   a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
   b. From the list of servers, click the name of your server.
   c. Select **Custom**.
   d. Select the **Runtime** tab. The following figure shows the Runtime tab.



Select	Counter	Type	Desc
□	BadRequests	CountStatistic	mySt
□	GoodRequests	CountStatistic	mySt
□	ResponseTime	TimeStatistic	mySt

Total 3

Figure 30. Runtime tab used for configuring PMI

e. Click **WBIStats.RootGroup**. This is a PMI sub module for data collected in the root group. This example uses the name WBIStats for the root group.
f. Click **ResourceAdapter**. This is a sub module for the data collected for the JCA adapters.
g. Click the name of your adapter, and select the processes you want to monitor.
h. In the right pane, select the check boxes for the statistics you want to gather, and then click **Enable**.

Results

PMI is configured for your adapter.

What to do next

Now you can view the performance statistics for your adapter.

Viewing performance statistics

You can view adapter performance data through the graphical monitoring tool, Tivoli Performance Viewer. Tivoli Performance Viewer is integrated with the administrative console in WebSphere Process Server.

Before you begin

Configure Performance Monitoring Infrastructure for your adapter.

Procedure

1. In the administrative console, expand **Monitoring and Tuning**, expand **Performance Viewer**, then select **Current Activity**.
2. In the list of servers, click the name of your server.
3. Under your server name, expand **Performance Modules**.

4. Click **WBIStatsRootGroup**.
5. Click **ResourceAdapter** and the name of your adapter module.
6. If there is more than one process, select the check boxes for the processes whose statistics you want to view.

Results

The statistics are displayed in the right panel. You can click **View Graph** to view a graph of the data, or **View Table** to see the statistics in a table format. The following figure shows adapter performance statistics as a graph.

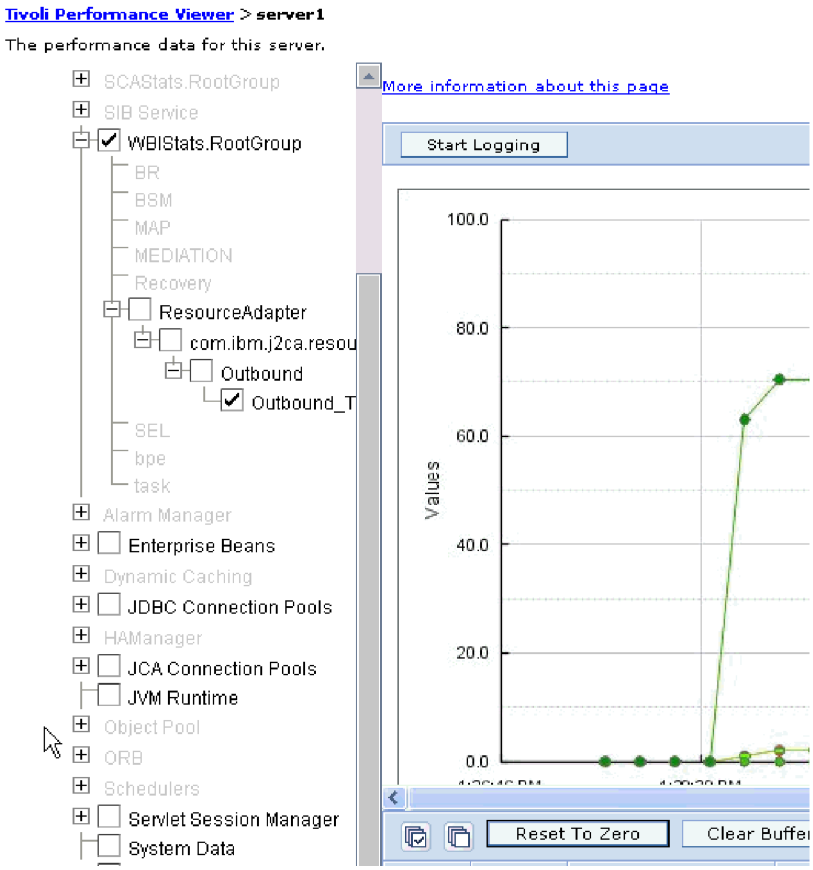


Figure 31. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure (CEI)

The adapter can use the Common Event Infrastructure, a component embedded in the server, to report data about critical business events such as the starting or stopping of a poll cycle. Event data can be written to a database or a trace log file depending on configuration settings.

About this task

Procedure

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 **Change Log Detail Levels** box, click the name of the CEI database (for example, WBIEventMonitor.CEI.ResourceAdapter.*) or the trace log file (for example, WBIEventMonitor.LOG.ResourceAdapter.*) to which you want the adapter to write event data.
5. Select the level of detail about business events that you want the adapter to write to the database or trace log file, and (optionally) adjust the granularity of detail associated with messages and traces.
   - **No Logging**. Turns off event logging.
   - **Messages Only**. The adapter reports an event.
   - **All Messages and Traces**. The adapter reports details about an event.
   - **Message and Trace Levels**. Settings for controlling the degree of detail the adapter reports about the business object payload associated with an event. If you want to adjust the detail level, choose one of the following:

     Fine. The adapter reports the event but none of the business object payload.

     Finer. The adapter reports the event and the business object payload description.

     Finest. The adapter reports the event and the entire business object payload.
6. Click **OK**.

Results

Event logging is enabled. You can view CEI entries in the trace log file or by using the Common Base Event Browser within the administrative console.

Chapter 8. Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly.

Support for the Log and Trace Analyzer

The adapter creates log and trace files that can be viewed with the Log and Trace Analyzer.

The Log and Trace Analyzer can filter log and trace files to isolate the messages and trace information for the adapter. It can also highlight the adapter's messages and trace information in the log viewer.

The adapter's component ID for filtering and highlighting is a string composed of the characters `ISESRA` plus the value of the adapter ID property. For example, if the adapter ID property is set to `001`, the component ID is `ISESRA001`.

If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for IBM i to `001` and `002`. The component IDs for those instances, `ISESRA001` and `ISESRA002`, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to `Instance01` and `Instance02`. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to `ISESRAInstanc`.

For outbound processing, the adapter ID property is located in both the resource adapter and managed connection factory property groups. If you update the adapter ID property after using the external service wizard to configure the adapter for outbound processing, be sure to set the resource adapter and managed connection factory properties consistently, to prevent inconsistent marking of the log and trace entries. For inbound processing, the adapter ID property is located only in the resource adapter properties, so this consideration does not apply.

For more information about the adapter ID property, see "Adapter ID (AdapterID)" on page 91. For more information about the Log and Trace Analyzer, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/topic/org.eclipse.hyades.log.ui.doc.user/concepts/cltaviews.htm.

Configuring logging and tracing

Configure logging and tracing to suit your requirements. Enable logging for the adapter to control the status of event processing. Change the adapter log and trace file names to separate them from other log and trace files.

About this task

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 takes 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.

Procedure

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. Click the names of 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.***.
   - For the Adapter for IBM i only, select the **com.ibm.j2ca.i5.*** package.
7. 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.
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.

Results

Log entries from this point forward contain the specified level of information for the selected adapter components.

Changing the log and trace file names

To keep the adapter log and trace information separate from other processes, use the administrative console to change the 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.

Before you begin

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

About this task

You can change the log and trace file names statically or dynamically. Static changes take effect when you start or restart the application server. Dynamic or run time 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.

Procedure

1. In the navigation pane of the administrative console, select **Applications > Enterprise Applications**.
2. In the Enterprise Applications list, click the name of the adapter application. This is the name of the EAR file for the adapter, but without the .ear file extension. For example, if the EAR file is named Accounting_OutboundApp.ear, then click **Accounting_OutboundApp**.
3. In the Configuration tab, in the Modules list, click **Manage Modules**.
4. In the list of modules, click IBM WebSphere Adapter for IBM i.
5. In the Configuration tab, under Additional Properties, click **Resource Adapter**.
6. In the Configuration tab, under Additional Properties, click **Custom properties**.
7. In the Custom Properties table, change the file names.
   a. Click either **logFilename** to change the name of the log file or **traceFilename** to change the name of the trace file.
   b. In the Configuration tab, type the new name in the **Value** field. By default, the log file is called SystemOut.log and the trace file is called trace.log.
   c. Click **Apply** or **OK**. Your changes are saved on your local machine.
   d. To save your changes to the master configuration on the server, use one of the following procedures:
      - **Static change**: Stop and restart the server. This method allows you to make changes, but those changes do not take effect until you stop and start the server.
      - **Dynamic change**: Click the **Save** link in the Messages box above the Custom properties table. Click **Save** again when prompted. This method allows you to make changes that take effect right away.

First-failure data capture (FFDC) support

The adapter supports first-failure data capture (FFDC), which provides persistent records of failures and significant software incidents that occur during run time in WebSphere Process Server or WebSphere Enterprise Service Bus.

The FFDC feature runs in the background and collects events and errors that occur at run time. The feature provides a means for associating failures to one another, allowing software to link the effects of a failure to their causes, and thereby facilitate the quick location of the root cause of a failure. The data that is captured can be used to identify exception processing that occurred during the adapter run time.

When a problem occurs, the adapter writes exception messages and context data to a log file, which is located in the *install_root*/profiles/*profile*/logs/ffdc directory.

For more information about first-failure data capture (FFDC), see the WebSphere Process Server or WebSphere Enterprise Service Bus documentation.

XAResourceNotAvailableException

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

Symptom:

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

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

Problem:

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

Solution:

To correct this problem, use the following procedure:

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

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

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

   server_install_directory\profiles*profile_name*\tranlog*node_name*\wps\ *server_name*\transaction\tranlog
3. Start the process server.

Self-help resources

Use the resources of IBM software support to get the most current support information, obtain technical documentation, download support tools and fixes, and avoid problems with WebSphere Adapters. The self-help resources also help you diagnose problems with the adapter and provide information about how to contact IBM software support.

Support Web site

The WebSphere Adapters software support Web site at http://www.ibm.com/software/integration/wbiadapters/support/ provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters, including the following types of

- Flashes (alerts about the product)
- Technical information including the product information center, manuals, IBM Redbooks®, and white papers
- Educational offerings
- Technotes

Recommended fixes

A list of recommended fixes you should apply is available at the following location: http://www.ibm.com/support/docview.wss?fdoc=aimadp&rs=695&uid=swg27010397

Technotes

Technotes provide the most current documentation about the Adapter for IBM i, including the following topics:
- Problems and their currently available solutions
- Answers to frequently asked questions
- How-to information about installing, configuring, using, and troubleshooting the adapter
- *IBM Software Support Handbook*

For a list of technotes for WebSphere Adapters, visit this address:

http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm

Plug-in for IBM Support Assistant

Adapter for IBM i provides a plug-in for IBM Support Assistant, which is a free, local software serviceability workbench. The plug-in supports the dynamic trace feature. For information about installing or using IBM Support Assistant, visit this address:

http://www.ibm.com/software/support/isa/

Business object error conditions

There are several known error conditions for the PutQueue, GetQueue, and CallPGM operations.

Known error conditions for the PutQueue operation

The following conditions are known to cause errors:
- An attempt is made to put a message on a data queue that is not existing
- An attempt is made to put a message on a data queue when the connection to IBM i host is broken

Known error conditions for the GetQueue operation

The following conditions are known to cause errors:
- An attempt is made to get a message from a data queue that is not existing
- An attempt is made to get a message from a data queue when the connection to IBM i host is broken

Known error conditions for the CallPGM operation

The following conditions are known to cause errors:

- An attempt is made to run a RPG program that is not existing
- An attempt is made to run an RPG program when the connection to the IBM i host is broken
- When not all ″in″ or ″inout″ attributes are populated with correct values
- When an invalid value is set for any of the parameters (for example, a string parameter is set in place of an integer parameter)
- When there are any exceptions or errors thrown from an underlying RPG program

Chapter 9. Reference

Detailed information about business objects, adapter properties (enterprise service discovery properties, resource adapter properties, managed (J2C) connection factory properties, activation specification properties, and interaction specification properties), messages, and related product information is provided for your reference.

Business object information

A business object is a structure that contains the data required by the adapter to perform an operation. The name of the business object is generated by the wizard in accordance with the naming convention for the adapter.

Business object attributes for operations

Business object attributes define the content of a business object which is simply a container for the data specified in the attributes. The structure of the data in the database is defined by the business object, but data in the database is in the business object attributes for operations.

Business object attributes for DataQueue operations

Table 4 lists the properties of a business object attributes for the DataQueue operations and describes their interpretation and settings.

Table 4. Attribute properties

Properties	Interpretation and settings
Maximum length of message	Maximum length for a message being added to an IBM i data queue recorded as an integer.
Message Content	The message content as a string.
Message Key	Indicates the key for a message on an IBM i data queue. **Note:** This is only for keyed data queues.
Name	This property represents the unique name of the attribute.
Type	This property specifies the type of the attribute, such as Integer or String.

Business object attributes for RPG operations

Table 5 lists the properties of a business object attribute and describes their interpretation and settings for running RPG programs on an IBM i system.

Table 5. RPG business object attribute properties

Properties	Interpretation and settings
type	Captures the data type of the attribute. The adapter uses the element name in the XPCML parameter list that corresponds to the PCML list, as the type. See Table 6 on page 86 for mapping between parameter type in the incoming XPCML and the generated type in business objects. The adapter internally generates the XPCML parameter from the PCML document that you select during external service discovery.

Table 5. RPG business object attribute properties (continued)

Properties	Interpretation and settings
passDirection	Determines whether an attribute is of nature `in` or `out` or `inout`. If a property is set as `out` or `inout` the adapter populates the result in that property. If it is set as `in`, the adapter does not populate it during a response.
totalDigits	The total number of digits in the field ('length' in PCML)
fractionDigits	The number of fractional digits ('precision' in PCML)
length	The length of the attribute. Applicable only for String types.
totalBytes	The field length in bytes ('length' in PCML).

Note: The adapter does not support 'structType' and array types parameters.

Table 6 lists the XPCML parameter type and the corresponding business object attributes.

Table 6. XPCML parameter type and business object attribute list

XPCML Parameter Type	Business object attribute type
shortParm	short
intParm	int
longParm	long
hexBinaryParm	hexBinary
floatParm	float
doubleParm	double
packedDecimalParm	decimal
structParm	Not applicable
stringParm	string
unsignedShortParm	unsignedShort
unsignedIntParm	unsignedInt
zonedDecimalParm	decimal

The adapter has the same set of limitations that the PCML standard does for a few parameter types. For more information about the PCML restrictions, see http://publib.boulder.ibm.com/infocenter/systems/index.jsp?topic=/rzasc/sc092507271.htm

Supported operations

Each business object is associated with an operation. The operation tells the adapter what to do with the business object. WebSphere Adapter for IBM i supports three different operations: GetQueue, PutQueue, and CallPGM.

In the external service wizard, you specify a name for the given operation. This name can be anything meaningful to you, such as GetMessageFromQueue or CallPGMbatch. The following table defines the operations that the adapter supports.

Table 7. Supported operations

Operation	Description
CallPGM	The CallPGM operation invokes an RPG program on an IBM i system
GetQueue	The GetQueue operation retrieves a message from an IBM i data queue
PutQueue	The PutQueue operation puts a message on an IBM i data queue

For detailed information on the operations supported by the Adapter for IBM i, see the topics devoted to it in this guide.

Naming conventions

When the external service wizard generates a business object, it provides a name for the business object that reflects the naming convention for the adapter. Typically, the business object name indicates the structure of the business object.

The adapter names business objects according to the <PREFIX> <Queue name> naming convention, where the prefix is DTAQ (for queue business objects) or PGM (for RPG program business objects) and the queue name is the actual name of the queue on the IBM i system. For example, the business object representing the QSYS.LIB/MYLIB.LIB/MYQUEUE.DTAQ data queue is named DTAQMYQUEUE by the adapter.

Business object names have no semantic value to the adapter or the IBM i system; that is, they derive no information nor meaning from the business object name. If one name is replaced by another, the adapter behavior remains the same.

You can rename business objects by using the refactoring functionality in WebSphere Integration Developer. For more details, refer to the WebSphere Integration Developer documentation.

The following table describes the naming conventions that the wizard uses for the business object.

Outbound configuration properties

WebSphere Adapter for IBM i has several categories of outbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and managed connection factory properties after you deploy the module to WebSphere Process Server or WebSphere Enterprise Service Bus using WebSphere Integration Developer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Guide to information about properties

The properties used to configure WebSphere Adapter for IBM i are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	A required field (property) must have a value in order for the adapter to work. Sometimes the external service wizard provides a default value for required properties. Removing a default value from a required field on the external service wizard *will not change that default value*. When a required field contains no value at all, the external service wizard will process the field using its assigned default value, and that default value will also be displayed on the administrative console. Possible values are **Yes** and **No**. Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example, • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.
Default	The predefined value that is set by the external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value. The word `None` is an acceptable default value, and does not mean that there is no default value.
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	Describes the property type. Valid property types include the following: • Boolean • String • Integer
Usage	Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented: For Rational® Application Developer for WebSphere Software version 6.40 or earlier, the password: • Must be uppercase • Must be 8 characters in length For versions of Rational Application Developer for WebSphere Software later than 6.40, the password: • Is not case sensitive • Can be up to 40 characters in length. This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.
Example	Provides sample property values, for example: "If Language is set to JA (Japanese), Codepage number is set to `8000`".
Globalized	If a property is globalized, it has national language support, meaning that you can set the value in your national language. Valid values are **Yes** and **No**.
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing pertains 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. Valid values are **Yes** and **No**.

Connection properties for the wizard

External service connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the IBM i system. These properties specify such things as connection configuration, bidirectional transformation properties, and logging options for the wizard. Once a connection is established, the wizard can discover in the IBM i server the metadata it needs to create business objects.

The following table lists and describes the connection properties. These properties can only be configured using the wizard and cannot be changed after deployment. A complete description of each property is provided in the sections that follow the table. For information about how to read the property details tables in the sections that follow, see “Guide to information about properties” on page 87.

Table 8. Connection properties for the external service wizard

Property name in the wizard	**Description**
User name	The IBM i system user name
Password	Password for the corresponding user name.
Host name	The fully qualified host name or IP address of the host where the is running
Object type to discover	Specifies the type of IBM i object the adapter will discover
“Path to folder on IBM i for object discovery” on page 90	Specifies the location of the folder on the IBM i server from where the discovery wizard should start discovering IBM i objects

User name (UserName)

This property specifies the user name for connecting to the IBM i system.

Table 9. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The user name for connecting to the IBM i system. Typing an incorrect user name will prevent the adapter from accessing the IBM i system.
Globalized	Yes
Bidi supported	Yes

Password (Password)

This property specifies the password for the IBM i system user name.

Table 10. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The password for the IBM i system user name. Typing an incorrect password will prevent the adapter from accessing the IBM i system.

Table 10. Password details (continued)

Globalized	Yes
Bidi supported	Yes

Host name (HostName)

This property specifies the IP address or fully qualified name of the host where the IBM i system is running.

Table 11. Host name details

Required	Yes
Default	No default value
Property type	String
Usage	Must be a valid server name
Example	user.us.ibm.com
Globalized	Yes
Bidi supported	Yes

Object type to discover

This property specifies the type of IBM i object the adapter will discover.

Table 12. Object type to discover details

Required	Yes
Default	No default value
Property type	String
Usage	Select one of the following options that best describes the types of business objects the adapter will discover for this module during outbound operation: • Data Queue • Report Program Generation
Restrictions	Only one type of object can be discovered at a time
Globalized	(Not available)
Bidi supported	(Not available)

Path to folder on IBM i for object discovery

This property specifies the folder on the IBM i server from where the discovery wizard should start discovering IBM i objects.

Table 13. Path to folder on IBM i for object discovery details

Required	Yes
Default	No default value
Property type	String
Usage	Value is not case sensitive
Example	/QSYS.LIB/PNPLIB.LIB/
Globalized	No

Table 13. Path to folder on IBM i for object discovery details (continued)

Bidi supported	No

Resource adapter properties

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

- LogFileMaxSize
- LogFileName
- LogNumberOfFiles
- TraceFileMaxSize
- TraceFileName
- TraceNumberOfFiles

The BusinessObjectNameSpace property has moved to the activation specification properties.

The following table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table. For information about how to read the property details table, see "Guide to information about properties" on page 87.

Table 14. Resource adapter properties for the Adapter for IBM i

Name		
In the wizard	**In the administrative console**	**Description**
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of X's instead of user data in log and trace files
(Not available)	enableHASupport	Do not change this property.
(Not available)	LogFileMaxSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
(Not available)	TraceFileMaxSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

This property identifies a specific deployment, or instance, of the adapter.

Table 15. Adapter ID details

Required	Yes

Table 15. Adapter ID details (continued)

Default	001
Property type	String
Usage	This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, ISESRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is ISESRA001. If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for IBM i to 001 and 002. The component IDs for those instances, ISESRA001 and ISESRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to ISESRAInstanc. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	Yes
Bidi supported	No

Disguise user data as ″XXX″ in log and trace files (HideConfidentialTrace) property

This property specifies whether to replace user data in log and trace files with a string of X’s to prevent unauthorized disclosure of potentially sensitive data.

Table 16. Disguise user data as ″XXX″ in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If you set this property to True, the adapter replaces user data with a string of X’s when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.

Table 16. Disguise user data as "XXX" in log and trace files details (continued)

Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to `true`.

Managed connection factory properties

Managed connection factory properties are used by the adapter at run time to create an outbound connection instance with the IBM i system.

You set managed connection factory properties through the external service wizard when you configure the adapter. You can change them before deployment through the WebSphere Integration Developer assembly editor or after deployment through the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

The following table describes the managed connection factory properties. More detailed information about each property is provided in tables that follow. For details about the information presented in these tables, see "Guide to information about properties" on page 87.

Note: The external service wizard refers to these properties as managed connection factory properties, while the administrative console refers to them as J2C connection factory properties.

Table 17. Managed connection factory properties for Adapter for IBM i

Property name		
In the wizard	**In the administrative console**	**Description**
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of X's instead of user data in log and trace files
Host name	HostName	The fully qualified host name or IP address of the host where the is running
Password	Password	Password for the corresponding user name.
User name	UserName	The IBM i system user name

Adapter ID (AdapterID)

This property identifies a specific deployment, or instance, of the adapter.

Table 18. Adapter ID details

Required	Yes
Default	`001`
Property type	String

Table 18. Adapter ID details (continued)

Usage	This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, `ISESRA`, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to `001`, the component ID is `ISESRA001`. If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for IBM i to `001` and `002`. The component IDs for those instances, `ISESRA001` and `ISESRA002`, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to `Instance01` and `Instance02`. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to `ISESRAInstanc`. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	Yes
Bidi supported	No

Disguise user data as ″XXX″ in log and trace files (HideConfidentialTrace) property

This property specifies whether to replace user data in log and trace files with a string of `X`'s to prevent unauthorized disclosure of potentially sensitive data.

Table 19. Disguise user data as ″XXX″ in log and trace files details

Required	No
Possible values	`True` `False`
Default	`False`
Property type	Boolean
Usage	If you set this property to `True`, the adapter replaces user data with a string of `X`'s when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	No
Bidi supported	No

Host name (HostName)

This property specifies the IP address or fully qualified name of the host where the IBM i system is running.

Table 20. Host name details

Required	Yes
Default	No default value
Property type	String
Usage	Must be a valid server name
Example	user.us.ibm.com
Globalized	Yes
Bidi supported	Yes

Password (Password)

This property specifies the password for the IBM i system user name.

Table 21. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The password for the IBM i system user name. Typing an incorrect password will prevent the adapter from accessing the IBM i system.
Globalized	Yes
Bidi supported	Yes

User name (UserName)

This property specifies the user name for connecting to the IBM i system.

Table 22. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The user name for connecting to the IBM i system. Typing an incorrect user name will prevent the adapter from accessing the IBM i system.
Globalized	Yes
Bidi supported	Yes

Interaction specification properties

Interaction specification properties control the interaction for an operation. The external service wizard sets the interaction specification properties when you configure the adapter. Typically, you do not need to change these properties. However, some properties for outbound operations such as encoding, object type to discover and so on can be changed. To change these properties after the application is deployed, use the assembly editor in WebSphere Integration Developer. The properties are located in the method binding of the import.

The following table lists the Interaction specification properties. More information about each property is provided in tables in the sections that follow. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 87.

Table 23. Interaction specification property for Adapter for IBM i

Property name	Description
Encoding	The type of character encoding to use
Object type to discover	Specifies the type of IBM i object the adapter will discover
Path to folder on IBM i for object discovery	Specifies the location of the folder on the IBM i system from where the external service wizard should start discovering IBM i objects

Encoding

This attribute is set during outbound communication to indicate the type of character encoding the adapter will use.

Table 24. Table 2. Encoding details

Required	No
Default	No default value
Attribute type	String
Usage	Encoding is used for headers, mail content, and attachment business objects.
Example	• ASCII for character encoding based on the English alphabet • Big5 for character encoding based on traditional Chinese characters • GB18030 for globalized characters in a business graph
Globalized	No
Bidi supported	No

Object type to discover

This property specifies the type of IBM i object the adapter will discover.

Table 25. Object type to discover details

Required	Yes
Default	No default value
Property type	String
Usage	Select one of the following options that best describes the types of business objects the adapter will discover for this module during outbound operation: • Data Queue • Report Program Generation
Restrictions	Only one type of object can be discovered at a time
Globalized	(Not available)
Bidi supported	(Not available)

Path to folder on IBM i for object discovery

This property specifies the folder on the IBM i system from where the discovery wizard should start discovering IBM i objects.

Table 26. Path to folder on IBM i for object discovery details

Required	Yes
Default	No default value
Property type	String
Usage	Value is not case sensitive
Example	/QSYS.LIB/PNPLIB.LIB/
Globalized	No
Bidi supported	No

Inbound configuration properties

WebSphere Adapter for IBM i has several categories of inbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using WebSphere Integration Developer or the administrative console, but connection properties for the external service wizard cannot be changed after deployment.

Guide to information about properties

The properties used to configure WebSphere Adapter for IBM i are described in detail in tables included in each of the configuration properties topics, such as Resource adapter properties, Managed connection factory properties, and so on. To help you use these tables, information about each row you might see is explained here.

The following table explains the meaning of each row that might be displayed in the table for a configuration property.

Row	Explanation
Required	A required field (property) must have a value in order for the adapter to work. Sometimes the external service wizard provides a default value for required properties. Removing a default value from a required field on the external service wizard *will not change that default value*. When a required field contains no value at all, the external service wizard will process the field using its assigned default value, and that default value will also be displayed on the administrative console. Possible values are **Yes** and **No**. Sometimes a property is required only when another property has a specific value. When this is the case, the table will note this dependency. For example, • Yes, when the EventQueryType property is set to Dynamic • Yes, for Oracle databases
Possible values	Lists and describes the possible values that you can select for the property.

Row	Explanation
Default	The predefined value that is set by the external service wizard. When the property is required, you must either accept the default value or specify one yourself. If a property has no default value, the table will state No default value. The word `None` is an acceptable default value, and does not mean that there is no default value.
Unit of measure	Specifies how the property is measured, for example in kilobytes or seconds.
Property type	Describes the property type. Valid property types include the following: • Boolean • String • Integer
Usage	Describes usage conditions or restrictions that might apply to the property. For instance, here is how a restriction would be documented: For Rational Application Developer for WebSphere Software version 6.40 or earlier, the password: • Must be uppercase • Must be 8 characters in length For versions of Rational Application Developer for WebSphere Software later than 6.40, the password: • Is not case sensitive • Can be up to 40 characters in length. This section lists other properties that affect this property or the properties that are affected by this property and describes the nature of the conditional relationship.
Example	Provides sample property values, for example: "If Language is set to JA (Japanese), Codepage number is set to `8000`".
Globalized	If a property is globalized, it has national language support, meaning that you can set the value in your national language. Valid values are **Yes** and **No**.
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing pertains 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. Valid values are **Yes** and **No**.

Connection properties for the wizard

External service connection properties are used to establish a connection between the external service wizard, a tool that is used to create business objects, and the IBM i system. These properties specify such things as connection configuration, bidirectional transformation properties, and logging options for the wizard. Once a connection is established, the wizard can discover in the IBM i server the metadata it needs to create business objects.

The following table lists and describes the connection properties. These properties can only be configured using the wizard and cannot be changed after deployment. A complete description of each property is provided in the sections that follow the table. For information about how to read the property details tables in the sections that follow, see "Guide to information about properties" on page 87.

Table 27. Connection properties for the external service wizard

Property name in the wizard	Description
User name	The IBM i system user name
Password	Password for the corresponding user name.
Host name	The fully qualified host name or IP address of the host where the is running
Object type to discover	Specifies the type of IBM i object the adapter will discover
"Path to folder on IBM i for object discovery" on page 100	Specifies the location of the folder on the IBM i server from where the discovery wizard should start discovering IBM i objects

User name (UserName)

This property specifies the user name for connecting to the IBM i system.

Table 28. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The user name for connecting to the IBM i system. Typing an incorrect user name will prevent the adapter from accessing the IBM i system.
Globalized	Yes
Bidi supported	Yes

Password (Password)

This property specifies the password for the IBM i system user name.

Table 29. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The password for the IBM i system user name. Typing an incorrect password will prevent the adapter from accessing the IBM i system.
Globalized	Yes
Bidi supported	Yes

Host name (HostName)

This property specifies the IP address or fully qualified name of the host where the IBM i system is running.

Table 30. Host name details

Required	Yes
Default	No default value
Property type	String
Usage	Must be a valid server name
Example	user.us.ibm.com

Table 30. Host name details (continued)

Globalized	Yes
Bidi supported	Yes

Object type to discover

This property specifies the type of IBM i object the adapter will discover.

Table 31. Object type to discover details

Required	Yes
Default	No default value
Property type	String
Usage	Select one of the following options that best describes the types of business objects the adapter will discover for this module during outbound operation: • Data Queue • Report Program Generation
Restrictions	Only one type of object can be discovered at a time
Globalized	(Not available)
Bidi supported	(Not available)

Path to folder on IBM i for object discovery

This property specifies the folder on the IBM i server from where the discovery wizard should start discovering IBM i objects.

Table 32. Path to folder on IBM i for object discovery details

Required	Yes
Default	No default value
Property type	String
Usage	Value is not case sensitive
Example	/QSYS.LIB/PNPLIB.LIB/
Globalized	No
Bidi supported	No

Resource adapter properties

The resource adapter properties control the general operation of the adapter, such as specifying the namespace for business objects. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

- LogFileMaxSize
- LogFileName
- LogNumberOfFiles
- TraceFileMaxSize
- TraceFileName

- TraceNumberOfFiles

The BusinessObjectNameSpace property has moved to the activation specification properties.

The following table lists the resource adapter properties and their purpose. A complete description of each property is provided in the sections that follow the table. For information about how to read the property details table, see “Guide to information about properties” on page 87.

Table 33. Resource adapter properties for the Adapter for IBM i

Name		
In the wizard	**In the administrative console**	**Description**
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing strings of X's instead of user data in log and trace files
(Not available)	enableHASupport	Do not change this property.
(Not available)	LogFileMaxSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
(Not available)	TraceFileMaxSize	Deprecated
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

This property identifies a specific deployment, or instance, of the adapter.

Table 34. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 34. Adapter ID details (continued)

Usage	This property identifies the adapter instance in log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, ISESRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is ISESRA001. If you run multiple instances of the same adapter, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose you set the adapter ID property of two instances of WebSphere Adapter for IBM i to 001 and 002. The component IDs for those instances, ISESRA001 and ISESRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately examine the log and trace information for each adapter instance because the component ID for both instances is truncated to ISESRAInstanc. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	Yes
Bidi supported	No

Disguise user data as ″XXX″ in log and trace files (HideConfidentialTrace) property

This property specifies whether to replace user data in log and trace files with a string of X's to prevent unauthorized disclosure of potentially sensitive data.

Table 35. Disguise user data as ″XXX″ in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, can be set both at the resource adapter level and the managed connection factory level. After using the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the WebSphere Integration Developer assembly editor or the administrative console to reset these properties, make sure you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

Do not change this property. It must be set to `true`.

Activation specification properties

Activation specification properties hold the inbound event processing configuration information for an export.

You set activation specification properties using the external service wizard during adapter configuration and change them before deployment through the WebSphere Integration Developer assembly editor or after deployment through the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

The following table summarizes the activation specification properties. A more complete description of each property is provided in the sections that follow the table. For information about how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 87.

Table 36. Activation specification properties for Adapter for IBM i

Property name		
In the wizard	**In the administrative console**	**Description**
Business object namespace	BusinessObjectNameSpace	The namespace for the business object definitions
Do not process events that have a timestamp in the future	FilterFutureEvents	Specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time
Ensure once-only event delivery	AssuredOnceDelivery	Specifies whether the adapter provides assured once delivery of events
Interval between polling periods	PollPeriod	The length of time that the adapter waits between polling periods
Failed event archiving control language program path	FailedEventArchivingControlLanguageProgramPath	The path to the control language program on the IBM i system that will be used to move failed events at run time
Failed events queue path	FailedEventsQueuePath	Path to the data queue on the IBM i system where failed events will be archived
Maximum connections	MaximumConnections	The maximum number of connections that the adapter can use for inbound event delivery
Minimum connections	MinimumConnections	The minimum number of connections that the adapter can use for inbound event delivery
Number of times to retry the system connection	RetryLimit	The number of times the adapter tries to reestablish an inbound connection after an error
Password	Password	Password for authorizing the user to retrieve events from the IBM i system

Table 36. Activation specification properties for Adapter for IBM i (continued)

Property name		
In the wizard	**In the administrative console**	**Description**
Poll quantity	PollQuantity	The number of events that the adapter delivers to the export during each poll period
Retry EIS connection on startup	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the IBM i system if it cannot connect at startup
Retry interval if connection fails	RetryInterval	The length of time that the adapter waits between attempts to establish a new connection after an error during inbound operations
Retry limit for failed events (FailedEventRetryLimit)	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed
Stop the adapter when an error is encountered while polling	StopPollingOnError	Specifies whether the adapter stops polling for events when it encounters an error during polling
Type of delivery	DeliveryType	Determines the order in which events are delivered by the adapter to the export
User name	UserName	The database user name to use for inbound events
"Polling queue " on page 112	PollingQueue	The absolute path to the queue in the IBM i server
Host name	HostName	The fully qualified host name or IP address of the host where the is running
Object type	ObjectType	Determines the object type that the adapter will poll
Key for keyed data queue	KeyForKeyedDataQueue	Fetches message from the keyed data queue (polling queue)
"Payload staging queue path (PayloadStagingQueuePath)" on page 113	PayloadStagingQueuePath	Specifies the absolute path of the queue that is used to store the message fetched from the poll queue for persistence
"Polling control language program path (PollingControlLanguageProgramPath)" on page 116	PollingControlLanguageProgramPath	Specifies an absolute path of the control language program that is used by the adapter to move the messages from the poll queue to the staging queue
Database schema name	EP_SchemaName	The schema name for the database used by the adapter's event persistence feature
Event recovery data source (JNDI) name	EP_DataSource_JNDIName	EP_DataSource_JNDIName
Event recovery table name	EP_TableName	The name of the event store used by the adapter
User name used to connect to event data source	EP_UserName	The user name used by the adapter's assured-once delivery feature.

Table 36. Activation specification properties for Adapter for IBM i (continued)

Property name		
In the wizard	**In the administrative console**	**Description**
Password used to connect to event data source	EP_Password	The password used by the adapter's event persistence feature. It is used to acquire the JDBC database connection from the data source configured in the server.

Business object namespace (BusinessObjectNameSpace)

This property specifies the namespace for the business object definitions.

Table 37. Business object namespace property characteristics

Required	No
Default	`http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi`
Property type	String
Usage	This value is added as a prefix to the business object name to keep business object names logically separated.
Example	The following example shows the Schema1Customer business object with the default namespace: `http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi/Schema1Customer`
Bidi supported	No

Data source JNDI name (DataSourceJNDIName)

This property specifies the name of the JNDI data source to use to establish a connection to the database.

Table 38. Data source JNDI name details

Required	No
Default	No default value
Property type	String

Table 38. Data source JNDI name details (continued)

Usage	Use this property to specify the JNDI name of a data source in WebSphere Process Server or WebSphere Enterprise Service Bus that specifies connection information for the target database. To improve the performance of inbound or outbound operations, specify the name of a data source that is enabled for prepared statement caching. If the user name and password properties are also set, they override the user name and password in the data source. The properties for connecting to the database are used in the following order: 1. If the DataSourceJNDIName property is set, the adapter uses it to establish the connection to the database. If the UserName and Password properties are also set, they override the user name and password set on the data source. 2. If the DataSourceJNDIName property is not set, the adapter uses the XADataSourceName and XADatabaseName properties, if set, to establish the connection. The DataSourceJNDIName property represents an XA or connection pool data source. If you define a JNDI data source on the server that supports XA transactions and then specify that data source when you configure the adapter, you can connect to any type of database that supports XA transactions. If you use an XA data source and database, the adapter supports XA transaction only for DB2® and Oracle databases. 3. If the DataSourceJNDIName, XADataSourceName, and XADatabaseName properties are not set, then the adapter uses the DatabaseURL, JDBCDriverClass, UserName and Password properties to establish the connection. Do not confuse the data source JNDI name property with the JNDI name of a managed connection factory or activation specification on the server. The following list highlights important differences between the types of JNDI names: • Data source JNDI name – Specifies a connection to a database – Used instead of saving user name and password in adapter properties – Saved as an adapter property • JNDI name of the managed connection factory or activation specification – Specifies a connection to a managed connection factory or activation specification on the server – Used instead of specifying the value of each managed connection factory or activation specification property in the wizard – Saved as the connection target in the import file
Globalized	Yes
Bidi supported	No

Type of delivery (DeliveryType)

This property specifies the order in which events are delivered by the adapter to the export.

Table 39. Type of delivery details

Required	No
Possible values	ORDERED UNORDERED
Default	ORDERED

Table 39. Type of delivery details (continued)

Property type	String
Usage	The following values are supported: • ORDERED: The adapter delivers events to the export one at a time. • UNORDERED: The adapter delivers all events to the export at once.
Globalized	No
Bidi supported	No

Do not process events that have a timestamp in the future (FilterFutureEvents)

This property specifies whether the adapter filters out future events by comparing the timestamp on each event with the system time.

Table 40. Do not process events that have a timestamp in the future details

Required	Yes
Possible values	True False
Default	False
Property type	Boolean
Usage	If set to True, the adapter compares the time of each event to the system time. If the event time is later than the system time, the event is not delivered. If set to False, the adapter delivers all events.
Globalized	No
Bidi supported	No

Ensure assured-only event delivery (AssuredOnceDelivery)

This property specifies whether to provide ensure once-only event delivery for inbound events.

Table 41. Ensure assured-only event delivery details

Required	Yes
Possible values	True False
Default	True
Property type	Boolean
Usage	When this property is set to True, the adapter provides assured once event delivery. This means that each event will be delivered once and only once. A value of False does not provide assured once event delivery, but provides better performance. When this property is set to True, the adapter attempts to store transaction (XID) information in the event store. If it is set to False, the adapter does not attempt to store the information. This property is used only if the export component is transactional. If it is not, no transaction can be used, regardless of the value of this property.
Globalized	No
Bidi supported	No

Event types to process (EventTypeFilter)

This property contains a delimited list of event types that indicates to the adapter which events it should deliver.

Table 42. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types
Default	`null`
Property type	String
Usage	Events are filtered by business object type. If the property is set, the adapter delivers only those events that are in the list. A value of `null` indicates that no filter will be applied and that all events will be delivered to the export.
Example	To receive only events relating to the Customer and Order business objects, specify this value: `Customer,Order`
Globalized	No
Bidi supported	No

Retry limit for failed events (FailedEventRetryLimit)

This property specifies the number of times that the adapter attempts to redeliver an event before marking the event as failed.

Table 43. Retry limit for failed events details

Required	No
Possible values	Integers
Default	`5`
Property type	Integer
Usage	Use this property to control how many times the adapter tries to send an event before marking it as failed. It accepts the following values: **Default** If this property is not set, the adapter tries five additional times before marking the event as failed. **0** The adapter tries to deliver the event an infinite number of times. When the property is set to `0`, the event remains in the event store and the event is never marked as failed. **> 0** For integers greater than zero, the adapter retries the specified number of times before marking the event as failed. **< 0** For negative integers, the adapter does not retry failed events.
Globalized	No
Bidi supported	No

Maximum connections (MaximumConnections)

This property specifies the maximum number of connections that the adapter can use for inbound event delivery.

Table 44. Maximum connections details

Required	No

Table 44. Maximum connections details (continued)

Default	1
Property type	Integer
Usage	Only positive values are valid. The adapter considers any positive entry less than 1 to be equal to 1. Typing a negative value or 1 for this property may result in runtime errors.
Globalized	No
Bidi supported	No

Minimum connections (MinimumConnections)

This property specifies the minimum number of connections that the adapter can use for inbound event delivery.

Table 45. Minimum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. Any value less than 1 is treated as 1 by the adapter. Typing a negative value or 1 for this property may result in run time errors.
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password for the IBM i system user name.

Table 46. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The password for the IBM i system user name. Typing an incorrect password will prevent the adapter from accessing the IBM i system.
Globalized	Yes
Bidi supported	Yes

Interval between polling periods (PollPeriod)

This property specifies the length of time that the adapter waits between polling periods.

Table 47. Interval between polling periods details

Required	Yes
Possible values	Integers greater than or equal to `0`.
Default	`2000`
Unit of measure	Milliseconds
Property type	Integer

Table 47. Interval between polling periods details (continued)

Usage	The poll period is established at a fixed rate, which means that if running the poll cycle is delayed for any reason (for example, if a prior poll cycle takes longer than expected to complete) the next poll cycle will occur immediately to make up for the lost time caused by the delay.
Globalized	No
Bidi supported	No

Maximum events in polling period (PollQuantity)

This property specifies the number of events that the adapter delivers to the export during each poll period.

Table 48. Maximum events in polling period details

Required	Yes
Default	`10`
Property type	Integer
Usage	The value must be greater than 0. If this value is increased, more events are processed per polling period and the adapter may perform less efficiently. If this value is decreased, fewer events are processed per polling period and the adapter's performance may improve slightly.
Globalized	No
Bidi supported	No

Number of times to retry the system connection (RetryLimit)

This property specifies the number of times the adapter tries to reestablish an inbound connection.

Table 49. Number of times to retry the system connection details

Required	No
Possible values	`0` and positive integers
Default	`0`
Property type	Integer
Usage	This property controls how many times the adapter retries the connection if the adapter cannot connect to the IBM i system to perform inbound processing. A value of `0` indicates an infinite number of retries. To control whether the adapter retries if it cannot connect to the IBM i system when it is first started, use the RetryConnectionOnStartup property.
Globalized	Yes
Bidi supported	No

Retry EIS connection on startup (RetryConnectionOnStartup)

This property controls whether the adapter attempts to connect again to the IBM i system if it cannot connect at startup.

Table 50. Retry EIS connection on startup details

Required	No

Table 50. Retry EIS connection on startup details (continued)

Possible values	`True` `False`
Default	`False`
Property type	Boolean
Usage	This property indicates whether the adapter should retry the connection to the IBM i system if the connection cannot be made when the adapter is started: • Set the property to `False` when you want immediate feedback about whether the adapter can establish a connection to the IBM i system, for example, when you are building and testing the application that receives events from the adapter. If the adapter cannot connect, the adapter writes log and trace information and stops. The administrative console shows the application status as `Stopped`. After you resolve the connection problem, start the adapter manually. • Set the property to `True` if you do not need immediate feedback about the connection. If the adapter cannot connect during startup, it writes log and trace information, and then attempts to reconnect, using the RetryInterval property to determine how frequently to retry and the value of the RetryLimit property to retry multiple times until that value is reached. The administrative console shows the application status as `Started`.
Globalized	No
Bidi supported	No

Retry interval if connection fails (RetryInterval)

When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to establish a new connection.

Table 51. Retry interval details

Required	Yes
Default	`2000`
Unit of measure	Milliseconds
Property type	Integer
Usage	Only positive values are valid. When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to establish a new connection.
Globalized	Yes
Bidi supported	No

Stop the adapter when an error is encountered while polling (StopPollingOnError)

This property specifies whether the adapter will stop polling for events when it encounters an error during polling.

Table 52. Stop the adapter when an error is encountered while polling details

Required	No
Possible values	`True` `False`
Default	`False`
Property type	Boolean

Table 52. Stop the adapter when an error is encountered while polling details (continued)

Usage	If this property is set to `True`, the adapter stops polling when it encounters an error. If this property is set to `False`, the adapter logs an exception when it encounters an error during polling and continues polling.
Globalized	No
Bidi supported	No

User name (UserName)

This property specifies the user name for connecting to the IBM i system.

Table 53. User name details

Required	Yes
Default	No default value
Property type	String
Usage	The user name for connecting to the IBM i system. Typing an incorrect user name will prevent the adapter from accessing the IBM i system.
Globalized	Yes
Bidi supported	Yes

Polling queue

This property specifies the absolute path to the queue in the IBM i system.

Table 54. Polling queue details

Required	Yes
Default	No default value
Property type	String
Usage	Must be a valid string value
Globalized	No
Bidi supported	No

Host name (HostName)

This property specifies the IP address or fully qualified name of the host where the IBM i system is running.

Table 55. Host name details

Required	Yes
Default	No default value
Property type	String
Usage	Must be a valid server name
Example	user.us.ibm.com
Globalized	Yes
Bidi supported	Yes

Object type (ObjectType)

This property determines the object type that the adapter polls.

Table 56. Object type details

Required	No
Default	No default value
Property type	String
Usage	Must be either a simple last-in first-out (LIFO) or first-in first-out (FIFO) data queue or a keyed data queue
Globalized	No
Bidi supported	No

Key value (KeyValue)

This property fetches message from the keyed data queue (polling queue).

Table 57. Key value details

Required	No
Default	No default value
Property type	String
Usage	Must be a valid string value
Globalized	No
Bidi supported	Yes

Payload staging queue path (PayloadStagingQueuePath)

This property specifies the absolute path of the queue that is used to store the message fetched from the poll queue for persistence.

Table 58. Payload staging queue path details

Required	Yes
Default	`None`No default value
Property type	String
Usage	Must be a valid string value. The payload staging queue is a keyed data queue that stores messages read from the polled queue.
Globalized	No
Bidi supported	No

Event recovery data source (JNDI) name property (EP_DataSource_JNDIName)

This property specifies the JNDI name of the data source that will be used by the event persistence class to acquire the JDBC database connection.

Table 59. Event recovery data source (JNDI) name details

Required	Yes, if the AssuredOnceDelivery property is set to `true`
Default	No default value

Table 59. Event recovery data source (JNDI) name details (continued)

Property type	String
Usage	This value is case-sensitive. The external service wizard does not create the data source on WebSphere Process Server or WebSphere Enterprise Service Bus. Make sure that the data source is defined on the server before deploying the adapter.
Globalized	Yes
Bidi supported	Yes

Event recovery table name property (EP_TableName)

This property specifies the name of the event store that will be used by the adapter for event persistence.

Table 60. Event recovery table name details

Required	Yes, if the AssuredOnceDelivery property is set to `true`
Default	
Property type	String
Usage	The value specified for this property must be unique for each adapter instance and cannot be used by another instance. This value is not case-sensitive.
Globalized	Yes
Bidi supported	Yes

Database schema name (EP_SchemaName)

This property specifies the schema name for the database used by the adapter's assured-once delivery feature.

Table 61. Database schema name details

Required	No
Default	No default value
Property type	String
Usage	Specifies the schema name for the database used by the adapter's assured-once delivery feature. This value is not case-sensitive.
Globalized	Yes
Bidi supported	Yes

Automatically create event table property (EP_CreateTable)

This property specifies whether the adapter should create an event store if it detects one does not exist.

Table 62. Automatically create event table details

Required	No
Possible values	`True` `False`
Default	`True`

Table 62. Automatically create event table details (continued)

Property type	Boolean
Usage	If set to True, the adapter will create an event store if it detects that one does not exist. If set to False, the adapter will not create an event store even if one does not already exist. Make sure that an event store exists if you want to ensure the delivery of events.
Globalized	No
Bidi supported	No

Password used to connect to event data source property (EP_Password)

This property specifies the password used by the adapter's assured-once delivery feature.

Table 63. Password used to connect to event data source details

Required	Yes, if the AssuredOnceDelivery property is set to `true`
Default	No default value
Property type	String
Usage	Specify the password that the adapter uses to connect to the event store. This value is case-sensitive.
Globalized	Yes
Bidi supported	Yes

User name used to connect to event data source property (EP_UserName)

This property specifies the user name used by the adapter's event assured-once delivery feature.

Table 64. User name used to connect to event data source details

Required	Yes, if you want assured-once delivery
Default	No default value
Property type	String
Usage	Specify the user name that the adapter uses to connect to the event store.
Globalized	Yes
Bidi supported	Yes

Failed events queue path (FailedEventsQueuePath)

This property specifies the path to the data queue on the IBM i system where failed events will be archived. Ensure that the failed events data queue is available on the IBM i system at run time.

Table 65. Failed events queue path details

Required	Yes
Default	`None`No default value
Property type	String

Table 65. Failed events queue path details (continued)

Usage	Must be a valid string value
Globalized	No
Bidi supported	No

Failed event archiving control language program path (FailedEventArchivingControlLanguageProgramPath)

This property specifies the path to the control language program on the IBM i system that will be used to move failed events at run time. Ensure that the control language program is available on the IBM i system at run time.

Table 66. Failed event archiving control language program path details

Required	Yes
Default	`None`No default value
Property type	String
Usage	Must be a valid string value
Globalized	No
Bidi supported	No

Polling control language program path (PollingControlLanguageProgramPath)

This property specifies an absolute path of the control language program that the adapter uses to move the messages from the poll quantity to the staging queue.

Table 67. Polling control language program path details

Required	Yes
Default	`None`No default value
Property type	String
Usage	The adapter moves poll quantity events at a time. As the staging queue is a keyed queue, the control language program also generates the unique key to write to the keyed data queue. In this way, the adapter assigns a unique identification number to each message it polled.
Globalized	No
Bidi supported	No

Globalization

WebSphere Adapter for IBM i is a globalized application that can be used in multiple linguistic and cultural environments. Based on character set support and the locale of the host server, the adapter delivers message text in the appropriate language. The adapter supports bidirectional script data transformation between integration components.

Globalization and bidirectional transformation

The adapter is globalized to support single- and multi-byte character sets and deliver message text in the specified language during outbound processing. 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

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, WebSphere Integration Developer, WebSphere Process Server, and WebSphere Enterprise Service Bus are written in Java. 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). Therefore, when data is transferred between these 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.

The adapter supports globalization for business objects and adapter foundation classes (AFC) related properties.

The inbound processing support provided by the adapter does not include polling of messages with globalized (non-English-language) characters from both keyed and simple data queues such as FIFO or LIFO.

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. There are multiple ways that a software application might display and process bidirectional script. WebSphere Process Server or WebSphere Enterprise Service Bus uses the Windows® standard format, but applications or file systems that exchange data with the server might use a different format. WebSphere Adapters transform bidirectional script data passed between the two systems so that it is accurately processed and displayed on both sides of a transaction.

Bidirectional format

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

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

Table 68. Bidirectional format attributes

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

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

Bidirectional properties that identify data for transformation

To identify business data that is subject to transformation, set the BiDiContext property. Do this by specifying values for each of the five bidirectional format attributes (listed in the preceding table) for the property.

Bidirectional transformation in business objects

The adapter supports bidirectional processing of the message content and key value of a business object.

Bidirectional formatting attributes of the interface

To enable bidirectional processing of business objects, you must modify the .import or .export file. For outbound processing, you must modify the business objects to enable the bidirectional transformation of the headers in the WebSphere Adapter for IBM i business object and the data in content-specific business objects.

The following example shows the biDiContext element that specifies the bidirectional transformation format string for the connection. Change the .import file for outbound modules and the .export file for inbound modules.

```
<connection ...>
 <properties>
  <biDiContext>VRYNN</biDiContext>
 </properties>
</connection>
```

Bidirectional formatting attributes of the business object

The following annotation, which contains the bidirectional context information, applies to the headers in the business object named fifoq, which is defined in the fifoq.xsd file. The adapter uses the bidirectional information in the BiDiContext element to transform the header values.

```
<complexType name="Fifoq">
<annotation xml:space="preserve">

<appinfo source="http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi/metadata">
 <ibmi:IBMiBusinessObjectTypeMetadata xsi:type="ibmi:IBMiBusinessObjectTypeMetadata"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:ibmi="http://www.ibm.com/xmlns/prod/websphere/j2ca/ibmi/metadata">
  <BiDiContext>
   <orientation>rtl</orientation>
   <textShape>nominal</textShape>
        <orderingScheme>visual</orderingScheme>
        <symmetricSwapping>true</symmetricSwapping>
        <numeralShapes>nominal</numeralShapes>
    </BiDiContext>
  </ibmi:IBMiBusinessObjectTypeMetadata>
</appinfo>

</annotation>
```

Properties enabled for bidirectional data transformation

WebSphere Adapter for IBM i has several configuration properties that are enabled for bidirectional data transformation.

The adapter enables the exchange of bidirectional data between a client application and the database, even if the data in the database is in a different bidirectional format than is used by the runtime environment. You can use bidirectional characters when configuring the adapter and in the application-specific information of your business objects. The following application specific information (ASI) is enabled for bidirectional support:

- Application specific information
  - Business object level ASI
  - Operation level ASI - ASI at the business object level is enabled for bidi support, but ASI at the operation levels is not supported in this release
  - Attribute level ASI - ASI at the business object level is enabled for bidi support, but ASI at the attribute levels is not supported in this release

The sections that follow list the ASI that is enabled for bidirectional transformation.

Bidi-enabled properties

The bidi-supported configuration properties for both inbound and outbound communication are shown in the following table. Only properties of type String are bidi-enabled. Integer properties, such as Port and Host, the Host name property, and Boolean type properties cannot be bidi-enabled.

Table 69. Bidi-enabled properties

In the wizard	In the administrative console
Message content	messageContent
Key value	KeyValue

Adapter messages

View the messages issued by WebSphere Adapter for IBM i at the following location.

Link to messages: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.wbit.620.help.messages.doc/messages.html

The displayed Web page shows a list of message prefixes. Click a message prefix to see all the messages with that prefix:

- Messages with the prefix CWYIS are issued by WebSphere Adapter for IBM i
- Messages with the prefix CWYBS are issued by the adapter foundation classes, which are used by all the adapters

Related information

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

Samples and tutorials

To help you use WebSphere Adapters, samples and tutorials are available from the Business Process Management Samples and Tutorials Web site. You can access the samples and tutorials in either of the following ways:

- From the welcome page that opens when you start WebSphere Integration Developer. To see samples and tutorials for WebSphere Adapter for IBM i, click **Retrieve**. Then browse the displayed categories to make your selections.
- At this location on the Web: http://publib.boulder.ibm.com/bpcsamp/index.html.

Information resources

- The WebSphere Business Process Management information resources Web page includes links to articles, Redbooks, documentation, and educational offerings to help you learn about WebSphere Adapters: http://www14.software.ibm.com/webapp/wsbroker/redirect?version=pix&product=wps-dist&topic=bpmroadmaps
- The WebSphere Adapters library page includes links to all versions of the documentation: http://www.ibm.com/software/integration/wbiadapters/library/infocenter/

Information about related products

- WebSphere Business Process Management, version 6.2, information center, which includes WebSphere Process Server, WebSphere Enterprise Service Bus, and WebSphere Integration Developer information: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/index.jsp

- WebSphere Adapters, version 6.1.x, information center: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/topic/com.ibm.wbit.612.help.adapter.emd.ui.doc/topics/tcreatecmps.html
- WebSphere Business Integration Adapters information center: http://publib.boulder.ibm.com/infocenter/wbihelp/v6rxmx/index.jsp?topic=/com.ibm.wbi_adapters.doc/welcome_adapters.htm

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

- WebSphere Adapters technical support: http://www.ibm.com/software/integration/wbiadapters/support/
- WebSphere Adapters technotes: http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm. In the **Product category** list, select the name of the adapter and click **Go**.

Notices

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

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

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

IBM Corporation
Department 2Z4A/SOM1
294 Route 100
Somers, NY 10589-0100
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: (c) (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. (c) 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, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

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

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

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

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

Printed in USA